AFRL-VA-WP-TR-2001-3053

AERODYNAMIC ANALYSIS FOR THE DESIGN ENVIRONMENT (AANDE)

VOLUME 3: PROGRAMMER'S MANUAL

M. H. LOVE
D. D. EGLE

LOCKHEED MARTIN TACTICAL AIRCRAFT SYSTEMS
AIRFRAME AND INSTALLATION
P.O. BOX 748, MAIL ZONE 2824
FORT WORTH, TX 76101-0748

NOVEMBER 1999

FINAL REPORT FOR PERIOD OF 30 SEPTEMBER 1995 – 30 JUNE 1998

Approved for public release; distribution unlimited.

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE UNITED STATES GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY BE RELATED TO THEM.

THIS REPORT IS RELEASEABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Victoria A. Tischler
VICTORIA A. TISCHLER
Aerospace Engineer
Structural Design and Development Branch

Terry M. Harris
TERRY M. HARRIS, Chief
Structural Design and Development Branch
Structures Division

DAVID M. PRATT, Technical Advisor
Structures Division
Air Vehicles Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	NOVEMBER 1999	Final, 09/30/1995 – 06/30/1998

4. TITLE AND SUBTITLE

AERODYNAMIC ANALYSIS FOR THE DESIGN ENVIRONMENT (AANDE)
VOLUME 3: PROGRAMMER'S MANUAL

5. FUNDING NUMBERS

C: F33615-95-C-3224
PE: 62201F
PR: 2401
TA: TI
WU: 05

6. AUTHOR(S)

M. H. LOVE
D. D. EGLE

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

LOCKHEED MARTIN TACTICAL AIRCRAFT SYSTEMS
AIRFRAME AND INSTALLATION
P.O. BOX 748, MAIL ZONE 2824
FORT WORTH, TX 76101-0748

8. PERFORMING ORGANIZATION REPORT NUMBER

FZM-8537

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
POC: Victoria A. Tischler, AFRL/VASD, (937) 255-9729

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-VA-WP-TR-2001-3053

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

This report is part of the documentations that describe the complete development of the "Aerodynamic ANalysis for the Design Environment (AANDE)." It is one of the three manuals that comprise the final report. The remaining reports consist of the AANDE Theoretical and Applications Study Manual (Volume 1) and the AANDE User's Manual (Volume 2).

The objective of the AANDE effort was to establish high quality reliable airloads for multidisciplinary design in ASTROS. This was accomplished by providing: a new steady linear aerodynamic procedure, alternate paths for the import of aerodynamic influence coefficient matrices and nonlinear pressure data, and a general asymmetric maneuver trim procedure.

The programmer's manual documents modifications to the ASTROS Programmer's Manual and includes a new section describing the format and use of the new PACKETS section.

14. SUBJECT TERMS

Aerodynamics, Analysis, Multidisciplinary Design, airloads, ASTROS, trim, asymmetric, PACKETS

15. NUMBER OF PAGES

168

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

ALPHABETICAL INDEX OF SOFTWARE MODULES

ALPHABETICAL INDEX OF DATABASE ENTITIES

FOREWORD

The Air Force Research Laboratory initiated development of the Automated Structural Optimization System, ASTROS, in 1983. Additional development work was conducted and completed in 1995 under contract F33615-87-C-3216. This document presents the analytical foundations for enhancements to ASTROS developed under the Aerodynamic Analysis for the Design Environment (AANDE) contract F33615-95-C-3224. This contract has been conducted by Lockheed Martin Tactical Aircraft Systems (LMTAS) and their subcontractor Universal Analytics Inc.. Lockheed Martin Aeronautical Systems also provided assistance to LMTAS in the AANDE program. Major contributors to the AANDE program include M.H. Love, the Program Manager, D.D. Egle, and D.K. Barker from LMTAS and R. Coopersmith from LMAS. From Universal Analytics, the major contributors were D.J. Neill, T. Shimko, S. Chen, and J. San Marco.

This report constitutes changes to the ASTROS version 12.0 Programmers Document (Ref. 5). It is one of four documents generated under the AANDE program.

Dr. Ray Kolonay was the primary Air Force program engineer for the AANDE program. Dr. V.B. Venkayya initiated the program and has provided much of the overall program direction.

1. INTRODUCTION

The unique attributes of ASTROS (Ref. 1) hold great potential for savings in design time, improvements in vehicle performance, and reductions in structural weight in aerospace vehicles. This potential has been limited due to capabilities lacking in modeling and simulation of maneuver loads for design. The overall objective of the Aerodynamic Analysis for the Design Environment contract (F33615-95-C-3224) is to establish high quality, reliable loads simulation in ASTROS. The Lockheed Martin Tactical Aircraft Systems team including Universal Analytics Inc. and Lockheed Martin Aeronautical Systems accomplished this objective by providing a new steady linear aerodynamic procedure, alternate paths for import of aerodynamic influence coefficient matrices and nonlinear pressure data, and a general asymmetric maneuver trim procedure.

The program encompassed three main tasks:

1.0 Phase I - System Specifications
2.0 Phase II - Module Development and Prototyping
3.0 Phase III - "Seamless" Integration and Verification.

In Phase I, changes to the ASTROS modules, paradigms, and data structures were identified, modeled, and tested against realistic scenarios of fighters, bombers, and transport aircraft. The results of these exercises formulated the plans for the software development and verification and are documented in the Software Design Guide (Ref. 2).

In Phase II, individual modules were developed and tested with realistic test cases as well as simple cases used for development. In Phase III, the modules were integrated into ASTROS through the memory manager, database, and MAPOL. Verification studies were performed simulating usage in a preliminary design scenario.

The AANDE contract is documented through four reports, Software Design Guide (Ref. 2), Programmer's Report (this report), the Theoretical and Application Studies Report (Ref. 3), and the User's Report (Ref. 4). The Software Design Guide (SDG) was developed at the end of Phase I of the AANDE program with the intention of defining requirements for ASTROS development beyond the original scope of AANDE but within the scope of air-loads in the design process. Many of the requirements in the SDG are implemented. The remaining documents are supplemental to the ASTROS version 12 documentation (Ref. 5). Areas of modification are documented fully. The Programmer's Report includes new and modified module descriptions (Section 2) and database (Section 3) entities. The User's Report includes new and modified ASTROS bulk data as well as user input requirements for QUADPAN.

THIS PAGE INTENTIONALLY LEFT BLANK.

2. SYSTEM INSTALLATION

Section 3.2 of the ASTROS Programmer's Manual (Reference 5) has been modified. It is in the following section, and new information has been underlined. Also a new section has been written describing the format and use of the new PACKETS section.

2.1 THE SYSTEM GENERATION PROGRAM

After development of the machine dependent source code for the target host machine, the next step in the ASTROS system installation is the assembly of the executable image of the ASTROS system generation program, SYSGEN. The libraries that must be linked to generate this program have been outlined in Section 2 of this manual while this section discusses the function of the SYSGEN program and details the structure of its inputs. These inputs not only define the standard ASTROS system but are also a powerful tool for an advanced user to expand the capabilities of the system. SYSGEN represents one of the most useful features of the ASTROS system architecture in that it provides for automated modification of many of the procedure's capabilities without requiring modification of any existing source code.

The purpose of SYSGEN is to create a system database (SYSDB) defining system parameters through the interpretation of several input files. Also, a FORTRAN routine is written by SYSGEN that provides the link between the ASTROS executive system and the application modules that comprise the run-time library of the procedure. This program unit is then linked with the system during the assembly of the ASTROS executable image. The resultant procedure makes use of the system database as a pool of data that defines the system at run time. These data are

1. the contents of the ASTROS run-time library of MAPOL addressable modules including both utility and application modules, usually delivered as MODDEF.DAT or MODDEF.DATA;

2. the ASTROS standard executive sequence composed of MAPOL source code statements, usually delivered as MAPOLSEQ.DAT or MAPOLSEQ.DATA;

3. the set of bulk data entries interpretable by the system and defined through the specification of bulk data templates to be interpreted by the ASTROS Input File Processor (IFP), usually delivered as TEMPLATE.DAT or TEMPLATE.DATA;

4. the set of relational schemata used by the executive system to satisfy the declaration of relational variables in the MAPOL sequence without forcing the user to explicitly define each schema at run time, usually delivered as RELATION.DAT or RELATION.DATA;

5. the set of input packet definitions, usually delivered as PACKET.DAT or PACKET.DATA. These describe the general format of the MAPOL, Edit, Solution Control, Function, Bulk Data, and QUADPAN input packets; and

6. the set of error message texts from which the UTMWRT system message writer utility builds error messages at run time, usually delivered as SERRMSG.DAT or SERRMSG.DATA.

2.2 INPUT PACKET DEFINITION

To allow for maximum generality in ASTROS input processing, there is a file called **PACKETS.DAT**. This file is organized as a sequence of specification entries that define different input

packets. There is one group of entries for each input packet. New packets may be defined by adding new groups of entries to this file. Naturally, new software will be required to read any such new packets.

The format of the PACKETS.DAT file is given in the following section.

2.2.1 The File Format

Each packet has the general form:

BEGINPACKET	*Input Packet Name*
STARTKEY	*Keyword*
ENDKEY	*Keyword*
FILENAME	*Filename*
CASE	**UPPER**, **LOWER** or none
BLANKLINE	**KEEP**, **MIXED** or none
ENDPACKET	

where:

STARTKEY	indicates the beginning of the packet.
ENDKEY	indicates the end of the packet.
FILENAME	is the name of the file where the input contents are saved.
BLANKLINE	is the instruction to indicate whether blank lines in the input packet will be kept or deleted.
CASE	indicates the letter case type to be used to convert the input packet.

Both the entries for each packet, and the inputs within each entry, may be in any order. The **CASE** and **BLANKLINE** records are optional. These packets are processed by SYSGEN and the rules defined by the entries are saved in a relation named PACKETRE on the system database. These are later used by the PREPAS module of ASTROS. Note that the **FILENAME** is used by PREPAS to define a relation where the packet data is saved.

2.2.2 The Standard ASTROS Packet Definition

The following is the **PACKETS.DAT** input for each of the standard system input data packets:

For the MAPOL Packet:

```
BEGINPACKET         MAPOL
        STARTKEY    MAPOL
        ENDKEY      ENDMAPOL
        FILENAME    MAPLPKT
        BLANKLINE   KEEP
ENDPACKET
```

For the EDIT Packet:

```
BEGINPACKET         EDIT
        STARTKEY    EDIT
        ENDKEY      ENDEDIT
        FILENAME    MAPLPKT
ENDPACKET
```

For the Solution Control Packet:

```
BEGINPACKET        SOLUTION
      STARTKEY     SOLUTION
      ENDKEY       ENDSOLUTION
      FILENAME     SOLNPKT
ENDPACKET
```

For the Function Packet:

```
BEGINPACKET        FUNCTION
      STARTKEY     FUNCTION
      ENDKEY       ENDFUNC
      FILENAME     FUNCPKT
ENDPACKET
```

For the Bulk Data Packet:

```
BEGINPACKET        BULKDATA
      STARTKEY     BEGIN BULK
      ENDKEY       ENDDATA
      FILENAME     BKDTPKT
ENDPACKET
```

For the QUADPAN Packet:

```
BEGINPACKET        QUADPAN
      STARTKEY     QUADPAN
      ENDKEY       ENDQUADPAN
      FILENAME     QDPANREL
      BLANKLINE    KEEP
ENDPACKET
```

2.3 MAPOL MODIFICATIONS

MAPOL was modified to support the longer entity names implemented in CADDB changes. This was done to facilitate the creation of "indexed" entity names in the aerodynamic model groups. Also, the CHARACTER variable type is supported for the MAPOL language.

Declaration	:	CHARACTER A, B, C;
Assignment	:	A := "string";
Comparison	:	IF A = "QUADPAN" THEN IF A "QUADPAN" THEN
Argument Passing	:	CALL MODULE (A);

CHARACTER data types may be set in a module and passed out to MAPOL with an updated value--just like other number variables.

Then, four new entity name types are created:

Table 1 New Entity Name Types For MAPOL Character Variables

<option>	Description
GGMEMBER	Group entity type
RGMEMBER	Relational group member
UGMEMBER	Unstructured group member
MGMEMBER	Matrix group member

These are basically entity name "variables" rather than entity name "symbols." The distinction is that the variables can be set in the module and passed through the MAPOL calling sequence. Regular entity names are static - once declared, they become a symbol rather than a variable name. The new feature is used in naming the members of a group. The GGMEMBER type is a RELATION on CADDB, but is denoted separately so that argument passing can perform the appropriate type checking. The other types are just like their non-group counterparts.

In MODDEF, these parameters are of type

CHARACTER	- 17
GGMEMBER	- 18
RGMEMBER	- 19
UGMEMBER	- 20
MGMEMBER	- 21

These values are specified on the parameter definition line in MODDEF.DAT (e.g., similar to RELATION which is a type 7 parameter).

3. ENGINEERING APPLICATIONS MODULES

This section presents updates to Chapter 5 of the ASTROS Programmer's Manual (Reference 5). Included are descriptions of new or modified Engineering Modules developed under this effort. These are summarized in the following table:

MODULE	DESCRIPTION
ACCFGEN	Initiate trim table of contents and assemble acceleration load parameters for static aeroelastic analysis.
AEROEFFS	Evaluates aeroelastic effectiveness sensitivities
AEROSENS	Computes the sensitivities of the rigid body accelerations and aerodynamic performance parameters (DCONTRM) for active steady aeroelastic subcases associated with the current subscript.
ARFMRG	To assemble the aerodynamic forces associated with the current subscript in the steady aeroelastic solution.
AROGNDRV	Examines solution control steady aerodynamic cases and create model group and member entity groups
AROHGEN	Generates the H matrix for creating a whole model from a half model in aeroelasticity. The H matrix will transform K, M or P matrices at F-set from structural basic coordinate system to aero coordinate system.
AROSNSDR	MAPOL director for saero sensitivity analyses
AROSNSMR	MAPOL director for saero sensitivity merge
AROSYMCK	Checks aerodynamic coordinate to decide whether a symmetry transformation is needed to create a full model from a user input half model. It also ensure that no nodes on the plane of symmetry is in the M-set.
DCEVAL	Evaluates displacement constraints in the current boundary condition
FLEXSTAB	Computes the rigid and flexible stability coefficients for steady aeroelastic analyses and the steady aeroelastic effectiveness constraints for constrained optimization steady aeroelastic analyses.
FLEXTRIM	To solve the trim equation for steady aeroelastic trim analyses.
FLXLODLD	Generates flexible trim parameter load vectors and to load group FLEXLOAD
FTRIMDRV	To examine and build groups for trim solver.
FTRIMOPT	To solve the trim equation for steady aeroelastic trim analyses while optimizing a user-defined objective function subject to user-defined constraints
GPIMPORT	Executes solution control aerodynamic model assembly command to import a model a group.
GRPARCHV	Executess solution control aerodynamic model assembly command for archiving groups.
LODSAGRP	Loads member groups and member entities into SAMODEL model group
LODSPGRP	To examine solution control steady aeroelastic cases and create SPLINE group and member entities.
MAKDFU	To assemble the sensitivities to the displacements of active stress and displacement constraints in the current active boundary condition.
MKDFSV	Calculates matrix [DFSV] which contains the S-matrix derivatives related to active stress/strain constraints
OFPAEROM	Solves for the static aero applied loads on the aero boxes and for the displacements on the aero boxes to satisfy the AIRDISP and TPRESSURE print/punch requests
OFPALOAD	Solves for the static aero applied loads and SPC forces to satisfy the print/punch requests. The resultant loads are written to the OGRIDLOD relation

OFPBMST	Calculate the trimmed BMST component load values, print out the results, and store the data on the **OBMSTLOD** relation
PARMBMST	Calculate the trim parameter BMST component loads data for use in trim optimization
PRETRM	Perform trim preface checks
QUADPAN	Processes the Quadrilateral Panel Method steady aerodynamics
RBMGEN	Generates G size rigid body mode matrix
RIGDSTAB	To Generate rigid steady aerodynamic stability derivative coefficients
RIGDTRIM	To solve the trim equation for rigid steady aerodynamic trim analyses.
RTRIMOPT	To solve the trim equation for rigid steady aerodynamic trim analyses while optimizing a user-defined objective function subject to user-defined constraints
SAERODRV	MAPOL director for steady aeroelastic analyses
SAEROMRG	Merges the static aero results for each subscript into the output matrix in case order rather than subscript order for the BCID'th boundary condition
SCEVAL	Computes the stress and/or strain constraint values for the statics or steady aeroelastic trim analyses in the current boundary condition
SCHDULER	To determine the value of scheduled aerodynamic parameters, provide the values to the trim modules, and check for convergence of the scheduled parameters.
SFORLD	To generate rigid structural trim parameter load vector group.
SOLUTION	To interpret the solution control packet
SPLINES	Generates the interpolation matrices that relate displacements and forces between the structural and steady aerodynamic MODELs.
SPLINFND	To obtain the names of spline matrices for the specified boundary condition .
SPLNGNDR	To examine solution control steady aeroelastic cases and create SPLINE group and member entities.
TRIMCHEK	To obtain the names of spline matrices for the specified boundary condition .
UDEFGEN	To assemble user defined load parameters for static aeroelastic analysis.
UDEFTRAN	To transform user defined load parameters of static aeroelastic analysis from centerline symmetric structural representation to full structural model in the case of applied substructuring techniques.
USSAERO	Processes the linear boundary condition steady aerodynamics

Engineering Application Module: **ACCFGEN**

Entry Point: **PACFGN**

Purpose:

Initiate trim table of contents and assemble acceleration load parameters for static aeroelastic analysis.

MAPOL Calling Sequence:

```
CALL ACCFGEN ( NITER, BCID, SUB, SYMTRN(BC), RIGDALOD, RIGDSLOD,
               STDYGEOM, TRIMDATA, CONLINK, TRIMTOC,
               [KFFX], TLABEL, [ACCFORCE], [ACCELOAD], MACH );
```

NITER	Optimization iteration number (Integer, Input)
BCID	Boundary condition number (Integer, Input)
SUB	Subscript value (Integer, Input)
SYMTRN(BC)	Logical denoting the presence of a "full" structural model instantiated from a centerline symmetric structural model via substructuring techniques (Input)
RIGDALOD	Group name and address of the current subscripted aerodynamic model rigid aerodynamic pressures (Input)
RIGDSLOD	Group name and address of the current subscripted aerodynamic model user defined rigid structural loads (Input)
STDYGEOM	Group name and address of the current subscripted aerodynamic model geometry (Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
CONLINK	Relation containing the **CONLINK** Bulk Data (Input)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aeroelastic loads matrices (Output)
[KFFX]	Matrix containing the structural stiffness in the F-set (Input)
TLABEL	Unstructured entity containing a master label list of trim control effectors and associated symmetries for current subscript. Used in downstream processes to assemble user defined loads and aerodynamic forces as well as acceleration forces in this routine (Output)
[ACCFORCE]	Matrix containing the null columns for acceleration loads in the structural domain (Output)
[ACCELOAD]	Matrix containing the null columns for acceleration loads in the aerodynamic domain (Output)

MACH Aerodynamic Mach number for current subscript (Real, Input)

Application Calling Sequence:

None

Method:

The module begins by bringing **TRIMDATA** data into memory associated with the current subscript maneuver trims. This information consists of the control parameters to be used in the ensuing cases. The **SUPORT** data is acquired to consolidate rigid body accelerations to be used in the maneuver trims. The **SUPORT** set ID is acquired from the **CASE** relation. Also, the trim symmetry is acquired from the **CASE** relation. The **CONLINK** data is retrieved for establishing the raw control parameters to be used in assembling the aeroelastic equation rigid loads.

The acceleration trim parameters and associated symmetries are assembled in a master table. Then a large loop is initiated to add the remaining unique trim parameters for all the trim cases in the current subscript. The loop is based on trim parameter labels attained from **TRIMDATA**. The loop first ascertains that the current label is not an acceleration parameter (i.e. already established). The logic then examines for the existence of the label as a **CONLINK** relation. If this is the case, the unique primary labels are acquired. The labels' associated load vectors are searched for in the current **RIGDALOD** aerodynamic and **RIGDSLOD** user defined load groups to assure their existence and obtain their symmetries. If found, the next label is acquired for search. If the label is not a **CONLINK** entity, then the label is determined to be a primary label and the search for its existence in the **RIGDALOD** and **RIGDSLOD** groups is executed. The routine terminates ASTROS execution if a load vector cannot be established for a trim label. The loop terminates after it has exhausted all of the labels from the **TRIMDATA** data sets.

Next, the sizes of the aerodynamic model and the structural model are determined by querying the **STDYGEOM** group and the **[KFFX]** matrix associatively. Also, the **TRIMTOC** relation is opened and prepared for loading. **TRIMTOC** will be the trim table of contents for the rigid load vectors and their associated labels for the current subscript.

A loop is initiated on the master label list. For every acceleration parameter a null load vector is created in the structural domain (**[ACCFORCE]**) and in the aerodynamic domain (**[ACCELOAD]**). One column for every acceleration parameter is created. For all the parameters, an entry in the **TRIMTOC** relation is created.

Finally, the unstructured entity, **TLABEL**, is created to aid in the downstream creation of load vector columns associated with the user defined loads and the aerodynamic loads. **TLABEL** contains three records. The first record is the number of labels. The second record is the symmetry of the trim subscript, and the third record is the list of labels in the same order in **TRIMTOC**.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **AEROEFFS**

Entry Point: **AROSEF**

Purpose:

Evaluates aeroelastic effectiveness sensitivities.

MAPOL Calling Sequence:

```
CALL AEROEFFS ( NITER, BCID, SUB, SYM, TRIMDATA, NDV, CONST, PCAE,
                [EFFSENS], [AMAT] );
```

NITER	Current iteration number (Input, Integer)
BCID	User defined boundary condition identification number. (Integer, Input)
SUB	Current static aeroelastic subscript number. (Input, Integer)
SYM	Symmetry flag for the current call. Either 1 for symmetric or -1 for antisymmetric. (Input, Integer)
TRIMDATA	Relation contains the TRIM bulk data and related boundary condition, subcase and subscript information. (Input)
NDV	Number of global design variables. (Input, Integer)
CONST	Relation of design constraints. (Input)
PCAE	Unstructured entity containing information indicating which pseudo-displacements (displacements due to unit configuration parameters) are active for the current design iteration. (Input)
EFFSENS	The matrix of dimensional stability derivative sensitivities. (Input)
AMAT	The matrix of constraint gradients. (Output)

Application Calling Sequence:

None

Method:

The **CASE** relation is read first to retrieve the **SUPPORT** set for the current boundary condition. The number and location of the support DOF are returned from the utility routine SEFCHK. Then the **CONST** relation is read for active lift effectiveness (**DCONCLA**), aileron effectiveness (**DCONALE**) and stability coefficient constraints (**DCONSCF**) for the current boundary condition, subscript and iteration.

The **EFFSENS** matrix, of dimension **NSUP*NDV*NAUE** where **NSUP** is the number of support dofs and **NAUE** is the number of active pseudo-displacement fields of the set computed in SAERO for the applied constraints.

The whole **EFFSENS** matrix is read into memory and then the loop over active constraints begins. For each active constraint, the **DISPCOL** attribute of the **CONST** relation is used to determine which

column of pseudo-displacements is associated with the constraint. The **PCAE** entity is then used to determine which column of the reduced set of active pseudo-displacement fields is the proper column. Once located, the constraint sensitivities may be computed from the dimensional stability coefficient derivatives and the normalization data stored in the **CONST** relation in the SAERO module. The constraint derivatives are computed from the following relationships.

The flexible stability coefficient response sensitivities which are required by the active user function constraints are also computed in this module. Those sensitivities are stored into relational and matrix entities to be used by the user function evaluation utilities.

Lift Effectiveness:

Upper Bound

$$\text{CLAREQ} > 0.0 \qquad DG/DX = \text{SENS ROW} / (\text{CLA}_{RIGID} * \text{CLAREQ})$$

$$\text{CLAREQ} < 0.0 \qquad DG/DX = -\text{SENS ROW} / (\text{CLA}_{RIGID} * \text{CLAREQ})$$

$$\text{CLAREQ} = 0.0 \qquad DG/DX = \text{SENS ROW} / \text{CLA}_{RIGID}$$

Lower Bound

$$\text{CLAREQ} > 0.0 \qquad DG/DX = -\text{SENS ROW} / (\text{CLA}_{RIGID} * \text{CLAREQ})$$

$$\text{CLAREQ} < 0.0 \qquad DG/DX = \text{SENS ROW} / (\text{CLA}_{RIGID} * \text{CLAREQ})$$

$$\text{CLAREQ} = 0.0 \qquad DG/DX = -\text{SENS ROW} / \text{CLA}_{RIGID}$$

where CLA_{RIGID} is stored in the **SENSPRM1** attribute of **CONST** and **CLAREQ** is stored in the **SENSPRM2** attribute of **CONST.**

Aileron Effectiveness:

Upper Bound

$$\text{AEREQ} > 0.0 \qquad DG/DX = (-\text{SENS}_1 * \text{CMXP}_{FLX} + \text{SENS}_2 * \text{CMXA}_{FLX}) / (\text{AEREQ} * \text{CMXP}_{FLX} **2)$$

$$\text{AEREQ} < 0.0 \qquad DG/DX = (\text{SENS}_1 * \text{CMXP}_{FLX} - \text{SENS}_2 * \text{CMXA}_{FLX}) / (\text{AEREQ} * \text{CMXP}_{FLX} **2)$$

$$\text{AEREQ} = 0.0 \qquad DG/DX = (-\text{SENS}_1 * \text{CMXP}_{FLX} + \text{SENS}_2 * \text{CMXA}_{FLX}) / (\text{CMXP}_{FLX} **2)$$

Lower Bound

$$\text{AEREQ} > 0.0 \qquad DG/DX = (\text{SENS}_1 * \text{CMXP}_{FLX} - \text{SENS}_2 * \text{CMXA}_{FLX}) / (\text{AEREQ} * \text{CMXP}_{FLX} **2)$$

$$\text{AEREQ} < 0.0 \qquad DG/DX = (-\text{SENS}_1 * \text{CMXP}_{FLX} + \text{SENS}_2 * \text{CMXA}_{FLX}) / (\text{AEREQ} * \text{CMXP}_{FLX} **2)$$

$$\text{AEREQ} = 0.0 \qquad DG/DX = (\text{SENS}_1 * \text{CMXP}_{FLX} - \text{SENS}_2 * \text{CMXA}_{FLX}) / (\text{CMXP}_{FLX} **2)$$

Stability Coefficient:

Upper Bound
```
REQ >0.0
   DG/DX = SENS ROW / REQ
REQ <0.0
   DG/DX = -SENS ROW / REQ
REQ = 0.0
   DG/DX = SENS ROW
```

Lower Bound
```
REQ >0.0
   DG/DX = -SENS ROW / REQ
REQ <0.0
   DG/DX = SENS ROW / REQ
REQ = 0.0
   DG/DX = -SENS ROW
```

where **REQ,** the dimensional required value is stored in the **SENSPRM1** attribute of **CONST**
The rows of **EFFSENS** associated with each constraint are dependent on the constraint type in the following way:

(1) Lift Effectiveness constraints always use the plunge DOF
(2) Aileron Effectiveness constraints always use the roll DOF
(3) Stability Coefficient constraints always use the row associated with the constrained axis. The constrained axis number (1,2,3,4,5,6) is stored in real form in the **SENSPRM2** attribute of **CONST**.

Design Requirements:
None

Error Conditions:
None

Engineering Application Module: **AEROSENS**

Entry Point: **AROSNS**

Purpose:

To compute the sensitivities of the rigid body accelerations and aerodynamic performance parameters (DCONTRM) for active steady aeroelastic subcases associated with the current subscript.

MAPOL Calling Sequence:

```
CALL AEROSENS ( NITER, BCID, MINDEX, SUB, CONST, SYM, NDV, BGPDT(BC),
                TRIMDATA, STABCFA, [PGAA], [LHSA(BC,SUB)],
                [RHSA(BC,SUB)], [DRHS], [AAR], [DDELDV], [AMAT] );
```

NITER	Current iteration number (Input, Integer)
BCID	User defined boundary condition identification number (Integer, Input)
MINDEX	Mach number index for the boundary condition to recover the proper stability coefficient data (Integer, Input)
SUB	The subscript identifier for the current SAERO subcases (Integer, Input)
CONST	Relation of constraint values (Character, Input)
NDV	The number of global design variables (Integer, Input)
SYM	The symmetry flag for the current SAERO subcases (Integer, Input)
BGPDT(BC)	Relation of basic grid point coordinate data (Character, Input), where BC represents the MAPOL boundary condition loop index number
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
STABCFA	Relation of rigid stability coefficient data (Character, Input)
[PGAA]	Partitioning vector used to obtain g-set active displacement and acceleration vectors for all static aero subcases that have active trim parameter, stress, strain and/or displacement constraints. (Input)
[LHSA(BC,SUB)]	Modified inertia matrix (Character, Input), where **BC** represents the MAPOL boundary condition loop index number
[RHSA(BC,SUB)]	Modified applied load matrix (Character, Input), where **BC** represents the MAPOL boundary condition loop index number
[DRHS]	Matrix entity containing the sensitivity of **[RHSA]** to the design variables (Character, Input)

[AAR]	Matrix entity containing the sensitivities of structural accelerations either zero (for fixed accelerations) or from solution of `LHSA*AAR = RHSA*DDELDV + DRHS (Output)`
[DDELDV]	Matrix entity containing the sensitivity of the configuration parameters to the design variables. Either zero (for FIXED control parameters) or from the solution of `LHSA*AAR = RHSA*DDELDV + DRHS (Output)`
[AMAT]	Matrix entity containing the sensitivities of the active aeroelastic control parameter (**DCONTRM**) constraints to the design variables (Character, Output)

Application Calling Sequence:

None

Method:

The minor modifications to the method have been made in the UAI Version 20.0 documentation for simplicity. The original document contains equations that are difficult to translate between word processing applications.

Design Requirements:

1. This module assumes that either strength and/or DCONTRM constraints exist for the static aeroelastic analyses in the current boundary condition.

Error Conditions:

None

Engineering Application Module: **ARFMRG**

Entry Point: **ARFMRG**

Purpose:

To assemble the aerodynamic forces associated with the current subscript in the steady aeroelastic solution.

MAPOL Calling Sequence:

```
CALL ARFMRG ( NITER, BCID, SUB, SYMTRN(BC), TRIMDATA, TLABEL,
              RIGDALOD, STDYGEOM, [HRGPTKF], [HIGPTKF], CASE,
              [AEROLOAD], [AIRFORCE], TRIMTOC, MACH, YESAERO );
```

NITER	Optimization iteration number (Integer, Input)
BCID	Boundary condition number (Integer, Input)
SUB	Subscript value (Integer, Input)
SYMTRN(BC)	Logical denoting the presence of a "full" structural model instantiated from a centerline symmetric structural model via substructuring techniques (Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TLABEL	Unstructured entity containing a master label list of trim control effectors and associated symmetries for current subscript. Used in downstream processes to assemble user defined loads and aerodynamic forces as well as acceleration forces in this routine (Input)
RIGDALOD	Group name and address of the current subscripted aerodynamic model rigid aerodynamic pressures (Input)
STDYGEOM	Group name and address of the current subscripted aerodynamic model geometry (Input)
[HRGPTKF]	Spline matrix to transform aerodynamic forces from the aerodynamic domain to aerodynamic forces in the structural domain in the F-set and in the case of substructuring for the real side of the centerline symmetric structure (Input)
[HIGPTKF]	Spline matrix to transform aerodynamic forces from the aerodynamic domain to aerodynamic forces in the structural domain in the F-set and in the case of substructuring for the image side of the centerline symmetric structure (Input)
CASE	Relation containing the case parameters for each analysis within each boundary condition as input in the solution control packet (Input)
[AEROLOAD]	Matrix containing the columns for aerodynamic loads in the aerodynamic domain. (Output)

`[AIRFORCE]`	Matrix containing the columns for aerodynamic loads in the structural domain in the G-set (Output)
`TRIMTOC`	Relation containing the Trim Table Of Contents which describes the aeroelastic load matrices (Input)
`MACH`	Aerodynamic Mach number for current subscript (Real, Input)
`YESAERO`	Logical indicating the presence of aerodynamic loads in the current subscript (Output)

Application Calling Sequence:

None

Method:

The maneuver trim symmetry is extracted from the **CASE** relation for the current boundary condition. The list of control parameters and associated aerodynamic symmetries for all of the trim maneuvers of the current boundary condition (and subscript by default) are extracted from the **TLABEL** unstructured entity. The size of the aerodynamic domain vectors is extracted from the **STDYGEOM** group, and the size of the structural domain vectors is extracted from the **[HRGPTKF]** matrix.

The logical variable **AEROSUB** is assigned based on the aerodynamic geometry and the value of the structural substructure variable **SYMTRN(BC)**. The aerodynamic geometric symmetry is maintained in the **STDYGEOM** relation. If the aerodynamic geometry is centerline symmetric and if **SYMTRN(BC)** is TRUE, then **AEROSUB** is set to TRUE. The ensuing logic will expect that the trim symmetry is asymmetric, and the **[AEROLOAD]** and **[AIRFORCE]** matrices will be assembled with symmetric and antisymmetric air loads column vectors intermingled. If however the **SYMTRN(BC)** is FALSE, either symmetric or antisymmetric **[AEROLOAD]** and **[AIRFORCE]** matrices will be assembled according to the trim symmetry. There is no provision for allowing assembly of an asymmetric solution for the case where the **SYMTRN(BC)** is FALSE and a centerline symmetric aerodynamic model (i.e. the routine will exit with a message to the user). If the aerodynamic geometry is not centerline symmetric, then **AEROSUB** is set to FALSE and ensuing logic requires that the trim symmetry be asymmetric. If **SYMTRN(BC)** is FALSE, then the logic expects that the spline matrix **[HRGPTKF]** is of the order of the full structural matrix, and the structural representation is a non-centerline symmetric geometry compatible with the aerodynamic geometry.

The aerodynamic load vectors are assembled in two passes. The first pass acquires basis loads, and the second pass acquires the loads for the remaining aerodynamic control parameters. The linear aeroelastic trim solver requires that the load vectors represent incremental perturbations in aerodynamic loads. Since the aerodynamic database maintains pressures at actual control parameter values (as opposed to incremental values from some basis), the perturbations must be constructed from the basis and actual pressure cases. The process is driven by the data from the **TLABEL** unstructured entity created in **ACCFGEN**.

The basis pressure vectors and associated **REFPARAM** relation addresses are retrieved from **RIGDALOD** for the symmetries identified in the creation of **TLABEL**. The **REFPARAM** entries contain the make-up of the basis pressure vector (i.e. control parameters used and set value at which the basis vector is valid). If **REFPARAM** is empty, then no basis vector exists. If a control parameter to be used in the construction of a trim case is not mentioned in the **REFPARAM**, its magnitude as a contributor to the basis vector is assumed to be 0. Usually, there is one basis vector. However, the case is provided for where there is both symmetric and antisymmetric basis vectors. This can happen

when the trim symmetry is asymmetric and both **AEROSUB** and **SYMTRN(BC)** are TRUE. The routine does not allow for multiple basis vectors in the same symmetry or any combination of asymmetric with symmetric or antisymmetric symmetries. For one symmetry, the routine only takes the first basis vector it comes to. For combinations of asymmetric with symmetric and antisymmetric, the routine will exit ASTROS and write a message to the user.

A single column load vector is constructed for the basis pressure vector (or each in the special case mentioned previously). If no basis exist, then a null column vector is created. Otherwise, the basis pressure vectors will be extracted into single column vectors. The control parameters and values from the **REFPARAM** relation are extracted for later use. The **[AEROLOAD]**, **[PF1]** (temporary) and **[AIRFORCE]** matrices are initiated. The structural vector length is determined from **[HRGPTKF]**. The **TRIMTOC** relation is readied for the addition of aerodynamic control parameter entries. It already has acceleration and user defined control parameters entries.

For the case of **AEROSUB** and **SYMTRN(BC)** being FALSE, the basis pressure vectors are read and packed in the **[AEROLOAD]** matrix. The basis pressure vector is operated on in the following equation to create the first column of the **[AIRFORCE]** matrix, and an entry is added to the **TRIMTOC** relation.

[PF1] = [HRGPTKF] *[PRFLOD]

For the case of **AEROSUB** being FALSE and **SYMTRN(BC)** being TRUE, the basis pressure vectors are read and packed in the **[AEROLOAD]** matrix. For this case, the spline matrix must have prior knowledge of the ordering of the structural matrix after it has been created through substructuring. For use with QUADPAN, the spline matrix is created in module SPLINE with this knowledge. The **[AEROLOAD]** matrix is operated on in the following equation to create the first column of the **[AIRFOCE]** matrix, and an entry is added to the **TRIMTOC** relation.

[PF1] = [HRGPTKF] *[PRFLOD]

For the case of **AEROSUB** being TRUE and **SYMTRN(BC)** being FALSE, the ARFMRG routine will provide a message to the user that this operation is not allowed, and ASTROS will exit.

For the case of **AEROSUB** being TRUE and **SYMTRN(BC)** being TRUE, the basis pressure vectors are read and packed in the **[AEROLOAD]** matrix. The **[AEROLOAD]** matrix is operated on in the following equation to create the first column of the **[AIRFORCE]** matrix, and an entry is added to the **TRIMTOC** relation.

[PF2] = [HRGPTKF] *[PRFLOD] + RSYM * [HIGPTKF] *[PRFLOD]

where for a symmetric parameter	**RSYM = 1**
and for an antisymmetric parameter	**RSYM = -1**

This process is now repeated for the incremental control parameter vectors. A loop is initiated over the parameter labels. The parameter's magnitude and pressure-vector-address are extracted from the unique **RIGDALOD** entries. The pressure vector is extracted from the database. An incremental pressure vector is created first by subtracting the basis pressure vector from the actual control parameter pressure vector. Similarly, the magnitude of the control parameter setting from the **REFPARAM** relation is subtracted from the magnitude of the control parameter setting of the actual pressures. The resulting difference in magnitude is divided into the difference of pressures to create the incremental pressures. If the pressure vector is a rate term, it is assumed to be normalized with respect to onset velocity. The routine multiplies the pressure vector by the onset velocity (also extracted from **RIGDALOD**) to provide units that other ASTROS routines expect downstream of

ARFMRG. The **[AEROLOAD]** is packed with the incremental pressure vector, and an entry is added to **TRIMTOC**.

For the case of **AEROSUB** and **SYMTRN(BC)** being FALSE, the incremental pressure vector is operated on in the following equation to create the first column of the **[AIRFORCE]** matrix.

[PF1] = [HRGPTKF] *[PRFLOD]

For the case of **AEROSUB** being FALSE and **SYMTRN(BC)** being TRUE, the incremental pressure vector is operated on in the following equation to create the first column of the **[AIRFORCE]** matrix. Recall in this case, the spline matrix must have full knowledge of the substructure operations that have occurred because the routine assumes the **[HRGPTKF]** matrix is non-square.

[PF1] = [HRGPTKF] *[PRFLOD]

For the case of **AEROSUB** being TRUE and **SYMTRN(BC)** being FALSE, the ARFMRG routine will provide a message to the user that this operation is not allowed, and ASTROS will exit.

For the case of **AEROSUB** being TRUE and **SYMTRN(BC)** being TRUE, the incremental pressure vector is operated on in the following equation to create the first column of the **[AIRFOCE]** matrix.

[PF2] = [HRGPTKF] *[PRFLOD] + RSYM * [HIGPTKF] *[PRFLOD]

where for a symmetric parameter	**RSYM = 1**
and for an antisymmetric parameter	**RSYM = -1**

Design Requirements:

The ACCFGEN module must have been called prior to the ARFMRG module.

Error Conditions:

None

Engineering Application Module: **AROGNDRV**

Entry Point: **AGENDR**

Purpose:

To examine solution control steady aerodynamic cases and create model group and member entity groups.

MAPOL Calling Sequence:

```
CALL AROGNDRV ( MINDEX, CASE, LOOP, GOAERO, CASEID, METHOD, MODEL,
                MACH, SYM, AICSYM, SAMODEL, STDYGEOM, RIGDALOD,
                AICMAT, AEROGRID, CAEROBOX, SACOMPS, SAGEOM, [AIC],
                [AAIC], [ASAIC], RIGDSLOD, NEWMODEL, SAROONLY );
```

MINDEX	Mach number index for the current pass. (Integer, Input)
CASE	Realtion contains solution control case definition. (Text, Input)
LOOP	Logical flag to indicate next pass is needed. (Logical, Output)
GOAERO	Logical flag to execute aero model generation. (Logical, Output)
CASEID	Subcase identification number. (Integer, Input)
METHOD	The name of the method which created the aero model. (Text, Output)
MODEL	Name of the current model. (Text, Output)
MACH	Mach number. (Real, Output)
SYM	Trim symmetry flag. (Integer, Output) = -1 Antisymmtric = 0 Asymmetric = 1 Symmetric
AICSYM	AIC matrix symmetry option. (Integer, Output) = -2 None = -1 Antisymmtric = 0 Asymmetric = 1 Symmetric = 2 Both symmetric and antisymmetric
SAMODEL	Group relation contains steady aerodynamic model group. (Text, Output)
STDYGEOM	Group relation contains steady aerodynamic geometry entities. (Text, Output)
RIGDALOD	Group relation contains rigid aerodynamic trim parameter load. (Text, Output)
AICMAT	Group relation contains AIC matrices. (Text, Output)

AEROGRID	Group member entity contains aerodynamic grid geometry relation. (Text, Output)
CAEROBOX	Group member entity contains aerodynamic connection relation. (Text, Output)
SACOMPS	Group member entity contains aerodynamic components relation. (Text, Output)
SAGEOM	Group member entity contains aerodynamic geometry relation. (Text, Output)
[AIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for **SYM**metric Mach numbers. (Text, Output)
[AAIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for **ANTISYM**metric Mach numbers. (Text, Output)
[ASAIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for **ASYM**metric Mach numbers. (Text, Output)
RIGDSLOD	Group relation contains rigid structural load. (Text, Output)
NEWMODEL	Logical flag indicating new model. (Logical, Output)
SAROONLY	Logical flag to indicate steady aero is the only discipline. (Logical, Output)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **SAERO** subcases in the current boundary condition. Then, for the specified input Mach index **MINDEX**, the boundary condition ID, subcase ID, aerodynamic model name, Mach number, the name of the method which creates the model, trim symmetry and AIC matrix symmetric option are obtained from **CASE** entries. The logical flag **NEWMDL** is also determined to indicate if it is a new model for the current subcase. If it is a new model and if the target model group is NOT assembled from solution control aerodynamic model assembly command, any non exist model group **SAMODEL**, related member groups and member entities are created.

If it is a new model and if the target model group is assembled from solution control aerodynamic model assembly command, the groups are fully or partially created by **IMPORT**, **ARCHIVE** or **OVERLAY** operations. Then, any related member groups and member entities are created if they are not created by solution control.
If the model itself exists, any related member groups and member entities are created if they do not exist.

Design Requirements:

This module must be called first before calling **USSAERO/QUADPAN** and **LODSAGRP** modules.

Error Conditions:

None

Engineering Application Module: **AROHGEN**

Entry Point: **AHGNDR**

Purpose:

To generate the H matrix for creating a whole model from a half model in aeroelasticity. The H matrix will transform K, M or P matrices at F-set from structural basic coordinate system to aero coordinate system.

MAPOL Calling Sequence:

```
CALL AROHGEN ( CASE, BGPDT(BC), USET(BC), RELES, BC, ACID, TOLVALUE,
               [HFREALT(BC)], [HFIMAGT(BC)], USETX(BC) );
```

CASE	Relation contains the case parameters for each subcases within each boundary condition. (Character, Input)
BGPDT(BC)	Relation of basic grid point coordinate data (Character, Input), where **BC** represents the MAPOL boundary condition loop index number.
USET(BC)	The unstructured entity defining structural sets(Character, Input)
RELES	Relation contains Bulk Data **RELES**. (Character, Input)
BC	the MAPOL boundary condition loop index number (Integer, Input)
ACID	The identification number for aerodynamic coordinate system. (Integer, Input)
TOLVALUE	The tolerance value to determine if the grid is on the aero symmetry plane. (Real, Input)
[HFREALT(BC)]	The H matrix for transforming real side K, M or P matrices at F-set (Character, Output)
[HFIMAGT(BC)]	The H matrix for transforming image side K, M or P matrices at F-set (Character, Output)
USETX(BC)	The unstructured entity defining updated structural sets (Character, Output)

Application Calling Sequence:

None

Method:

This module begins by bringing relation **BGPDT** into memory and sorting the entries via internal ID. Then, relation **RELES** entries and unstructured entity **USET** are brought into memory. All the grid points in **BGPDT** are checked to determined whether a symmetric transformation is needed to create a full model from user input half model. After obtaining the transformation matrix from aerodynamic coordinate system to basic coordinate system, a 6X6 H matrix for the real side is generated from that aero to basic transformation matrix, while a 6X6 H matrix for the image side is generated from multiplying that aero to basic transformation matrix with symmetric transformation matrix. Finally, the F-size H matrices for real side and image side are generated by looping over all grid points in **BGPDT**

entries, checking grid set information with **USET**, and packing the corresponding terms from 6X6 H matrices.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **AROSNSDR**

Entry Point: **AROSDR**

Purpose:

MAPOL director for saero sensitivity analyses.

MAPOL Calling Sequence:

```
CALL AROSNSDR ( NITER, BCID, SUB, NDV, LOOP, MINDEX, CONST, SYM,
                TRIMDATA, NGDR,  [PGDRG(BC)], [UAG(BC)], [AAG(BC)],
                ACTUAG, [UGA], [AGA], [PGAA], [PGAU], PCAA, PRAA,
                [UAGC(BC,SUB)], [AAGC(BC,SUB)],  ACTAEFF, [AUAGC],
                [AAAGC], PCAE, [UAGI(BC)], [AAGI(BC)], [UGAI],
                [AGAI],[UAGCI(BC,SUB)], [AAGCI(BC,SUB)], [AUAGCI],
                [AAAGCI] );
```

NITER	Current iteration number (Input, Integer)
BCID	User defined boundary condition identification number (Integer, Input)
SUB	Current static aeroelastic subscript number (Input, Integer)
NDV	Number of design variables (Input, Integer)
LOOP	A logical flag set to indicate whether additional **MINDEX** subscripts are needed to complete the processing of all the active Mach number/Symmetry conditions on all the **TRIM** entries. One pass for each unique active Mach number will be performed with **MINDEX** set as appropriate for the active pass until this routine returns **LOOP=FALSE** (Logical, Output)
MINDEX	Mach number index value of the current pass (Output, Integer)
CONST	Relation of design constraints (Input)
SYM	Symmetry flag for the current pass. Either 1 for symmetric or -1 for antisymmetric (Output, Integer)
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
NGDR	Denotes dynamic reduction in the boundary condition (Input, Integer) **0** No **GDR** **1** **GDR** is used
[PGDRG(BC)]	A partitioning vector that removes the additional GDR scalar points from the g-set sized displacement and acceleration vectors. Required only if **NGDR** $\neq$ **0** (Input), where BC represents the MAPOL boundary condition loop index number

[UAG(BC)] g-set displacement vector for all static aero subcases in the current boundary condition (Input), where **BC** represents the MAPOL boundary condition loop index number

[UAGI(BC)] Image side g-set displacement vector for all static aero subcases in the current boundary condition for substructure reflection (Input), where **BC** represents the MAPOL boundary condition loop index number

[AAG(BC)] g-set acceleration vector for all static aero subcases in the current boundary condition (Input), where **BC** represents the MAPOL boundary condition loop index number

[AAGI(BC)] Image side g-set acceleration vector for all static aero subcases in the current boundary condition for substructure reflection (Input), where **BC** represents the MAPOL boundary condition loop index number

ACTUAG Logical flag that is set to TRUE if there are any active constraints that require the displacements or accelerations. Those constraints are trim parameters, stresses, strains and displacements (Output, Logical)

[UGA] Reduced g-set active displacement vectors for all static aero subcases that have active trim parameter, stress, strain and/or displacement constraints. This is a subset of the columns of **[UAG(BC)]** and does not include the GDR scalar points, if any (Output)

[UGAI] Image side reduced g-set active displacement vectors for substructure reflection for all static aero subcases that have active trim parameter, stress, strain and/or displacement constraints. This is a subset of the columns of **[UAG(BC)]** and does not include the GDR scalar points, if any (Output)

[AGA] Reduced g-set active acceleration vectors for all static aero subcases that have active trim parameter, stress, strain and/or displacement constraints. This is a subset of the columns of **[AAG(BC)]** and does not include the GDR scalar points, if any (Output)

[AGAI] Image side reduced g-set active acceleration vectors for substructure reflection for all static aero subcases that have active trim parameter, stress, strain and/or displacement constraints. This is a subset of the columns of **[AAG(BC)]** and does not include the GDR scalar points, if any (Output)

[PGAA] Partitioning vector used to obtain **[UGA]** and **[AGA]** from **[UAG(BC)]** and **[AAG(BC)]** (Output)

[PGAU] Partitioning vector relative to **[UAG(BC)]** and **[AAG(BC)]** that marks the displacement/acceleration columns associated with subcases having active stress, strain or displacement constraints. This vector will be identical to [PGAA] unless there are subcases in which **DCONTRM** constraints are active and no stress, strain or displacement constraints are active (Output)

PCAA	An unstructured entity with one word for each active stress, strain or displacement constraint in the current subscript related subcases. That word is the subcase number associated with the constraint (Output)
PRAA	An unstructured entity with one word for each element stress, strain or displacement response function required by the active user function constraints in the current subscript related subcases. That word is the subcase number associated with the response (Character,Output)
[UAGC(BC,SUB)]	g-set pseudo-displacement vectors (displacement fields due to loads arising from unit values of trim configuration parameters) for all aeroelastic effectiveness constraints (Input), where BC represents the MAPOL boundary condition loop index number
[UAGCI(BC,SUB)]	Image side g-set pseudo-displacement vectors (displacement fields due to loads arising from unit values of trim configuration parameters) for all aeroelastic effectiveness constraints for substructure reflection (Input), where BC represents the MAPOL boundary condition loop index number
[AAGC(BC,SUB)]	g-set pseudo-acceleration vectors (acceleration fields due to loads arising from unit values of trim configuration parameters) for all aeroelastic effectiveness constraints (Input), where BC represents the MAPOL boundary condition loop index number
[AAGCI(BC,SUB)]	Image side g-set pseudo-acceleration vectors (acceleration fields due to loads arising from unit values of trim configuration parameters) for all aeroelastic effectiveness constraints for substructure reflection (Input), where BC represents the MAPOL boundary condition loop index number
ACTAEFF	Logical flag that is set to TRUE if there are any active constraints that require the pseudo-displacements or pseudo-accelerations. Those constraints are **DCONALE, DCONCLA** and **DCONSCF** (Output, Logical)
[AUAGC]	Reduced g-set active pseudo-displacement vectors for all active effectiveness constraints. This is a subset of the columns of **[UAGC(BC)]** and does not include the GDR scalar points, if any (Output)
[AUAGCI]	Image side reduced g-set active pseudo-displacement vectors for all active effectiveness constraints for substructure reflection. This is a subset of the columns of **[UAGCI(BC)]** and does not include the GDR scalar points, if any (Output)
[AAAGC]	Reduced g-set active pseudo-acceleration vectors for all active effectiveness constraints. This is a subset of the columns of **[AAGC(BC)]** and does not include the GDR scalar points, if any (Output)

[AAAGCI]	Image side reduced g-set active pseudo-acceleration vectors for all active effectiveness constraints for substructure reflection. This is a subset of the columns of **[AAGCI(BC)]** and does not include the GDR scalar points, if any (Output)
PCAE	An unstructured entity with one word for each active effectiveness constraint (**DCONALE, DCONCLA, DCONSCF**) in the current subscript's related subcases. That word is the column id of the first column associated with the constraint (Output)

Application Calling Sequence:

None

Method:

This module treats two distinct families of aeroelastic constraints for the current boundary condition and subscript number: the active aeroelastic effectiveness constraints **DCONALE, DCONCLA** and **DCONSCF**; and the active displacement dependent constraints **DCONTRM, DCONDSP**, stress and strain. Two parallel sets of partitioning operations take place to extract the active pseudo-displacements needed for effectiveness constraints and active displacements needed for the displacement-dependent constraints. The control information for the presence or absence of each type of constraint and the additional control information to extract data from downstream entities is also prepared for each constraint family. Finally, the need to loop through another subscript value is determined and the **LOOP** variable is output. **LOOP** will be false after the last needed **AROSNSDR** call for the current **BC**.

First **CASE** is queried to obtain the **TRIM** identification number and symmetry. Then **TRIM** is read to obtain the subscript numbers, **MINDEX** values and subcase ids for each **SAERO** subcase in the current **BC**. These data are then assembled into a master table containing the trim identification number, the subscript number and the subcase id.

The **CONST** relation is then read to count the number of active stress, strain, displacement, aileron effectiveness, lift effectiveness, stability coefficient and trim parameter constraints. A loop over each **CONST** entry is then made to assemble the partitioning vectors and control information for sensitivity computations. Each family of constraints is treated separately.

For effectiveness constraints, the **DISPCOL** attribute in **CONST** is used to build a partitioning vector for the active pseudo-displacements and accelerations. The partitioning vector is later destroyed but the active column numbers are stored as a contiguous string of numbers and written to **PCAE.** For lift effectiveness constraints there is one **UAGC/AAGC** column for each applied constraint: the disp/accel. due to a unit angle of attack. For aileron effectiveness, there are two columns: the first due to unit control surface deflection and the second due to unit roll rate. For stability coefficients, there is one column due to a unit deflection of the constrained parameter. As the constraints are looped over, only those with the current subscript value are considered. Those with lower subscript values have already been processed and, if any active constraints are found with a higher subscript value, the **LOOP** flag is set to TRUE to ensure another pass is done.

A similar path exists for the displacement-dependent constraints except the matrices being partitioned are the actual displacement and acceleration fields. Separate partitioning vectors are assembled for 1) active columns due to all displacement dependent constraints (**PGAA**) and 2) active columns due to stress, strain and displacement constraints (**PGUA**). Again, previously processed subscripts are ignored and **LOOP** is set to true if larger subscripts are encountered.

Finally, the assembled partitioning vectors are written to their respective entities and the **PCAE** and **PCAA** entities are determined from the partitioning data and written to the unstructured entities. The

presence of active constraints in the effectiveness family or displacement-dependent family is then known and the **ACTAEFF** and **ACTUAG** flags, respectively, are set.

The element stress and strain responses; displacement responses; aeroelastic flexible stability coefficent responses; and trim parameter responses which are required by active user functional constraints at the current boundary condition and subscript number are treated in the similar manner as those corresponding constraints. The subcases which have active displacement or element stress/strain response functions are also defined as active. The partitioning vector, **PGAA**, and the set of subcase numbers that are active, **PRAA**, are loaded if necessary. The **ACTAEFF** and **ACTUAG** flags are also set for active responses.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **AROSNSMR**

Entry Point: **AROSMR**

Purpose:
MAPOL director for saero sensitivity analyses.

MAPOL Calling Sequence:

```
CALL AROSNSMR ( BCID, SUB, TRIMDATA, NDV, [PGAA], [PGAU], [MATOUT],
                [MATSUB] );
```

BCID	User defined boundary condition identification number (Integer, Input)
SUB	Current static aeroelastic subscript number (Input, Integer)
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
NDV	Number of design variables (Input,Integer)
[PGAA]	Partitioning vector used denoting active displacement fields for the current boundary's static aeroelastic subcases (Input)
[PGAU]	Partitioning vector used denoting active displacement fields that are active due only to stress, strain and displacement constraints for the current boundary's static aeroelastic subcases (Input)
[MATOUT]	On input, **MATOUT** must contain the merged, reordered displacement or acceleration sensitivities for all the subcases processed for the earlier subscript values. On output the **SUB**'th subscript is included. This matrix will contain one column for each active vector for the 1st design variable, followed by another set for the second and so on. The order of the vectors within each variable's set will be the order of the **SAERO** subcases in the **CASE** relation (Input and Output)
[MATSUB]	The input matrix of displacement or acceleration sensitivities for all the subcases processed for the **SUB**'th subscript. This matrix will contain one column for each active vector associated with the SUB'th subscript for the 1st design variable, followed by another set for the second and so on. The order of the vectors within each variable's set will be the order of the **TRIM** ids appearing in the **TRIM** relation associated with the **SUB**'th subscript value (Input)

Application Calling Sequence:
None

Method:
First the **CASE** relation is read to retrieve the trim id's for the **SAERO** subcases in the current boundary condition. The the **TRIM** relation is read to obtain the subcase numbers associated with each trim id having the current **SUB**script value. Then the **PGAA** and **PGUA** vectors are read into memory to assist in the partitioning operation.

Then the **MATSUB** and **MATOUT** matrices are opened. If **MATOUT** is uninitialized or if **SUB** = 1, it is initialized (flushed and the number of rows, precision and form set to those of **MATSUB**. If **MATOUT** already exists and has data in it, a scratch matrix is created to hold the final merged data.For each design variable in the model, each **SAERO CASE** entry for the current boundary is processed. For each **CASE** entry, the partitioning vector **PGAA** is used to determine if it is active and therefore may have a column in either **MATSUB** or **MATOUT**. For the active subcase id, the **TRIM** data are searched to determine the subscript number associated with the subcase. If the subscript is less than **SUB**, a column from **MATOUT** may be taken (if it was stored there on an earlier pass). If the subscript is equal to **SUB**, it may be stored on the output matrix from **MATSUB**. If greater than **SUB**, it is ignored till later passes.

Once a column is identified as active in **MATSUB** (**PGAA** indicates active and subscript = **SUB**), an additional check is made to see if the column is active in **PGUA**. Only those columns that are active in **PGUA** are copied to **MATOUT**. This filtering is done to limit the amount of computational effort in the stress, strain and displacement constraint sensitivity computations that proceed using the **MATOUT** matrix. The **MATSUB** columns that are active due to **DCONTRM** constraints are no longer needed as these sensitivities are assumed to have been computed already in the **AEROSENS** module.

Once the final matrix is formed, if **MATOUT** had had data in it, the name of the scratch matrix that was loaded is switched with that of **MATOUT**. The scratch entity is then destroyed.

Design Requirements:

1. The assumption is that each **MATSUB** matrix contains the results from the "**SUB**"th subscript value in the order of the trim id's for that **SUB** appear in the **TRIM** relation.

2. The same **MATOUT** matrix must be passed into the **AROSNSMR** module on each call since the columns associated with earlier subscript values are read from **MATOUT** into a scratch entity. The merged matrix that results then replaces the input **MATOUT**.

3. The **AEROSENS** module is called upstream of the **AROSNSMR** module to process active **DCONTRM** constraints for the current subscript. Thus, those columns that are active only for **DCONTRM** constraints may be filtered out for the downstream processing of stress, strain and displacement constraints.

Error Conditions:

None

Engineering Application Module: **AROSYMCK**

Entry Point: **ACHKDR**

Purpose:

This module does aerodynamic coordinate checking to decide whether a symmetry transformation is needed to create a full model from a user input half model. It also ensure that no nodes on the plane of symmetry is in the m-set.

MAPOL Calling Sequence:

```
CALL AROSYMCK ( CASE, BGPDT(BC), USET(BC), RELES, BC, ACID, TOLVALUE,
                STRSYM );
```

CASE	Relation contains the case parameters for each subcases within each boundary condition. (Character, Input)
BGPDT(BC)	Relation of basic grid point coordinate data (Character, Input), where BC represents the MAPOL boundary condition loop index number.
USET(BC)	The unstructured entity defining structural sets(Character, Input)
RELES	Relation contains Bulk Data **RELES**. (Character, Input)
BC	the MAPOL boundary condition loop index number (Integer, Input)
ACID	The identification number for aerodynamic coordinate system. (Integer, Input)
TOLVALUE	The tolerance value to determine if the grid is on the aero symmetry plane. (Real, Input)
STRSYM	The logical flag to indicate if the structure needs to be reflected into a full model (Logical, Output).

Application Calling Sequence:

None

Method:

This module begins by bringing relation **BGPDT** into memory and sorting the entries by internal ID. Then, relation **RELES** entries and unstructured entity **USET** are brought into memory. All the grid points in **BGPDT** are checked to determined whether a symmetric transformation is needed to create a full model from user input half model.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **DCEVAL**

Entry Point: **DCEVAL**

Purpose:

To evaluate displacement constraints in the current boundary condition.

MAPOL Calling Sequence:

```
CALL DCEVAL ( NITER, BCID, [UG(BC)], BGPDT(BC), CONST, BSAERO,
              TRIMDATA, [UAGI(BC)] );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
[UG(BC)]	Matrix of displacement vectors in the g-set for the boundary condition (Input), where BC represents the MAPOL boundary condition loop index number.
[UGI(BC)]	Image side matrix of displacement vectors in the g-set for the boundary condition for steady aerodynamic substructure reflection (Input), where **BC** represents the MAPOL boundary condition loop index number.
BGPDT(BC)	Relation of basic grid point coordinate data (Character, Input), where **BC** represents the MAPOL boundary condition loop index number.
CONST	Relation of constraint values (Character, Input)
BSAERO	Static aeroelastic flag; =1 if this call is associated with static aeroelastic analyses. (Optional, Integer, Input)
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)

Application Calling Sequence:

None

Method:

The module first determines if there are any **DCONST** options for a STATIC (**BSAERO=0**) or SAERO (**BSAERO=1**) discipline for the current boundary condition and terminates if there are none. If there are, a loop is made through all the subcases for the current boundary condition and the necessary displacement constraint(s) are calculated and written to the **CONST** relation. Finally, the displacement responses which are required by any user functional constraints are computed.

Design Requirements:

1. This module appears within the analysis portion of the OPTIMIZE segment of the MAPOL sequence. It is within the analysis boundary condition loop and must follow the recovery of the displacement vector to the g-set.

Error Conditions:

None

Engineering Application Module: **FLEXSTAB**

Entry Point: **FLXSTB**

Purpose:

To compute the rigid and flexible stability coefficients for steady aeroelastic analyses and the steady aeroelastic effectiveness constraints for constrained optimization steady aeroelastic analyses.

MAPOL Calling Sequence:

```
CALL FLEXSTAB ( NITER, BCID, SUB, SYM, QDP, TRIMDATA,
                STABCFA, STABCFS, BGPDT(BC), [LHSA(BC,SUB)],
                [RHSA(BC,SUB)], [AAR], [DELTA(SUB)],
                [PRIGID], [R33], CONST, AEFLG(SUB), [AARC]
                [DELC], SYMTRN(BC), STDYGEOM, DOTRMCON );
```

NITER	Optimization iteration number (Integer, Input)
BCID	Boundary condition number (Integer, Input)
SUB	Subscript number of subcases considered in this call (Integer, Input)
SYM	The symmetry flag for the current subcases (Integer, Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
STABCFA	Relation of aerodynamic rigid stability coefficient data in the aerodynamic domain (Input)
STABCFS	Relation of aerodynamic stability coefficient data (rigid and flexible) in the structural domain (Output)
BGPDT(BC)	Relation of basic grid point coordinate data, where BC represents the MAPOL boundary condition loop index number (Input)
[LHSA(BC,SUB)]	Matrix of modified inertia coefficients (Input), where **BC** represents the MAPOL boundary condition loop index number
[RHSA(BC,SUB)]	Matrix of applied load vectors reduced to the r-set (Input), where **BC** represents the MAPOL boundary condition loop index number
[AAR]	Matrix of acceleration vectors (Output)
[DELTA(SUB)]	Matrix of configuration parameters (Output)
[PRIGID]	Rigid Load Matrix (Input)
[R33]	Reduced rigid body mass matrix (Input)

SYMTRN(BC)	Logical denoting the presence of a "full" structural model instantiated from a centerline symmetric structural model via substructuring techniques (Input)
STDYGEOM	Group name and address of the current subscripted aerodynamic model geometry (Input)
DOTRMCON	Logical denoting whether to evaluate trim constraints in module **ARCOND** (Input)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **FLEXTRIM** subcases in the current boundary condition. Then, the **STABCFA** relation is read into memory. Aerodynamic reference data from **AEROS** are read into memory. The aerodynamic model symmetry is acquired from **STDYGEOM**. From this, the logical **FULAERO** is set to allow proper computation of stability derivatives. The logical **FULLMODL** is then set based on **SYMTRN(BC)** and **FULAERO**. The **TRIMDATA** relations are read for all entries that have the current subscript value. **CONEFFS** and **CONLINK** data are also read into memory.

Then an evaluation of the trim data is done to determine the number of trim subcases that will be solved during this pass (for the current subscript). The **SUPCHK** utility is used to evaluate the **SUPORT** conditions to ensure (again) that is satisfies the requirements of the **TRIM** solver and to get the names and DOFs of the supported degrees of freedom. Then, after creating needed scratch entities, the grand loop on the trim subcases begins.

Stability coefficients for each trim subcase must be solved separately because of the options for control effectiveness and control linking. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** ID of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **FLEXSTAB** entries in **CASE** for each **BC**).

Then the unrestrained stability coefficients are computed followed by the restrained stability coefficients. The **FTSCFS/D** submodule is called for the unrestrained stability derivatives. The data is computed at the support degrees of freedom due to the unit configurations:

$$[F] = [MRR][LHS]^{-1}[RHS]$$

The **PRIGID** matrix contains the rigid aerodynamic data in the structural domain. These rigid and flexible data are normalized and stored in the **STABCFS** relation with the relational qualifiers "RSPLINED" for the rigid splined derivatives and "UNRESTRA" for the flexible unrestrained derivatives. Once complete, the stability coefficient table is printed using the effectiveness parameters and linking terms to assemble the "dependent" coefficients and factor all coefficients according to the user input.

Using the table of derivatives, the **ARCONS/D** submodule is called to evaluate the constraints for the current subcase. These constraints are evaluated from the stability coefficient table but, to prepare for when eventual sensitivity computations, the additional outputs **AEFLG**, **AARC**, and **DELC** matrices are needed. The first is a logical flag to indicate to the MAPOL sequence that the **AARC** and **DELC** matrices are full. The **AARC** matrix and **DELC** matrix contain one or more columns for each constraint (appended in the order the constraints are evaluated). The **AARC** contains the accelerations

of the support DOFs due to the unit configuration parameter vectors in **DELC**. This pair of matrices will allow the computation of the derivative of the accelerations due to the unit parameters which is an essential ingredient in the sensitivity computation.

For lift effectiveness constraints:

AARC - 1 column due to unit **ALPHA**
DELC - 1 column containing a unit **ALPHA** with all others 0.0

For aileron effectiveness constraints:

AARC - 2 columns; the first for unit **SURFACE** rotation and the second for unit roll rate (**PRATE**).
DELC - 2 columns containing a unit rotation of the named **SURFACE** and the second a unit **PRATE**

For stability coefficient constraints (DCONSCF):

AARC - 1 column due to unit **PARAMETER** where **PARAMETER** is that named on the constraint entry
DELC - 1 column containing a unit **PARAMETER** with all others 0.0

Then the restrained stability coefficients are computed followed by the restrained stability coefficients. The **RSSCFS/D** submodule is called for the restrained stability derivatives. The data is computed at the support degrees of freedom due to the unit configurations:

$$[F] = [RHS]$$

The **PRIGID** matrix contains the rigid aerodynamic data in the structural domain. These rigid and flexible data are normalized and stored in the **STABCFS** relation with the relational qualifiers "RSPLINED" for the rigid splined derivatives and "RESTRAIN" for the flexible unrestrained derivatives. Once complete, the stability coefficient table is printed using the effectiveness parameters and linking terms to assemble the "dependent" coefficients and factor all coefficients according to the user input.

Using the table of derivatives, the **ARCONS/D** submodule is called to evaluate the constraints for the current subcase as in the **FTSCFS/D** submodule.

DCONTRM are not evaluated at this time.

Design Requirements:
None

Error Conditions:
None

Engineering Application Module: **FLEXTRIM**

Entry Point: **FLXTRM**

Purpose:

To solve the trim equation for steady aeroelastic trim analyses.

MAPOL Calling Sequence:

```
CALL FLEXTRIM ( NITER, BCID, SUB, SYM, QDP, TRIMDATA, TRIMRSLT,
                TRIMTOC, [LHSA(BC,SUB)], [RHSA(BC,SUB)], [AAR],
                [DELTA(SUB)], [PRIGID], [R33] );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **FLEXTRIM** subcases considered in this call (Integer, Input)
SYM	The symmetry flag for the current **FLEXTRIM** subcases (Integer, Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMRSLT	Relation containing **TRIMDATA** and results of the trim solutions (Output)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
[LHSA(BC,SUB)]	Matrix of modified inertia coefficients (Input), where **BC** represents the MAPOL boundary condition loop index number
[RHSA(BC,SUB)]	Matrix of applied load vectors reduced to the r-set (Input), where **BC** represents the MAPOL boundary condition loop index number
[AAR]	Matrix of acceleration vectors (Output)
[DELTA(SUB)]	Matrix of configuration parameters (Output)
[PRIGID]	Rigid Load Matrix (Input)
[R33]	Reduced rigid body mass matrix (Input)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **FLEXTRIM** subcases in the current boundary condition. Then, the **TRIMTOC** relation is read into memory. The **TRIMDATA** and **TRIMRSLT** relations are read for all entries that have the current subscript value. Other trim data from **AEROS**, **CONEFFS**, and **CONLINK** are also read into memory.

Then an evaluation of the trim data is done to determine the number of trim subcases that will be solved during this pass (for the current subscript). The **SUPCHK** utility is used to evaluate the **SUPORT** conditions to ensure (again) that is satisfies the requirements of the **TRIM** solver and to get the names and DOFs of the supported degrees of freedom. Then, after creating needed scratch entities, the grand loop on the trim subcases begins.

Each trim subcase must be solved separately because of the options for control effectiveness and control linking. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** id of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **FLEXTRIM** entries in **CASE** for each **BC**). Then, the **AROLNK** routine is called to assemble a linking matrix of control effectiveness factors and linking relationships for the current subscript such that:

$$\{\delta\} = [\mathbf{TLINK}] * \mathbf{DELRED}$$

where the **DELRED** matrix is reduced to only the active trim parameters and the effectiveness factors have been included. Then the rigid and flexible loads are hit with the linking matrix to reduce the problem to the relevant configuration parameters:

$$\mathbf{P2RED} = \mathbf{P2} * \mathbf{TLINK}$$

$$\mathbf{RHSRED} = \mathbf{RHS} * \mathbf{TLINK}$$

P2RED and **RHSRED** contain one row for each structural acceleration and one column for each non-acceleration label on the trim entry. This means that the total number of stability parameters (either fixed or free) is the number of columns in **P2** and **RHS**. Further, the order of the parameters is the order given on the **TRIM** tuples.

Now the trim equations can be assembled. From the input, we have the relationship

$$\begin{bmatrix} LHS_{ff} & LHS_{fk} \\ LHS_{kf} & LHS_{kk} \end{bmatrix} \begin{bmatrix} AR_{free} \\ AR_{known} \end{bmatrix} = \begin{bmatrix} RHS_{fu} & RHS_{fs} \\ RHS_{ku} & RHS_{ks} \end{bmatrix} \begin{bmatrix} DEL_u \\ DEL_s \end{bmatrix}$$

Where:	Represents:
F+K	Number of **SUPORT** point DOFs
F	Set of free accelerations, **AR**
K	Set of known **(FIXED)** accelerations, **AR**
U+S	Number of **AERO** parameters
U	Set of unknown parameters
S	Set of set **(FIXED)** parameters

These equations must be rearranged to get free accelerations and unknown deltas on the same side of the equation:

$$\begin{bmatrix} LHS_{ff} & -RHS_{fu} \\ LHS_{uf} & -RHS_{uu} \end{bmatrix} \begin{bmatrix} AR_{free} \\ DEL_u \end{bmatrix} = \begin{bmatrix} -LHS_{kk} & RHS_{ks} \\ -LHS_{sk} & RHS_{ss} \end{bmatrix} \begin{bmatrix} AR_k \\ DEL_s \end{bmatrix}$$

and we must handle the degenerate case where all accelerations or all deltas are known.

Following rearrangement of the equations, the unknowns are solved for in the **ARTRMS/D** routine. First the rigid masses and loads, **P2RED** and **MRR**, are used to obtain the rigid trim and then the flexible inputs, **RHSRED** and **LHS**, are used for the "real" solution.

The flexible results are unscrambled and the rigid body accelerations (either input on the **TRIM** entry or output from the solution of the above) are stored on the **AAR** matrix and the same is done with the trim parameters after the **TLINK** matrix is used to recover the full vector from the reduced set. Then the results for the rigid and flexible trim are printed.

The module repeats the entire process for all the subcases that are associated with the current **SUB**script and then terminates.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **FLXLODLD**

Entry Point: **FLXLOD**

Purpose:

To generate flexible trim parameter load vectors and to load group FLEXLOAD.

MAPOL Calling Sequence:

```
CALL FLXLODLD ( BCID, SUB, TRIMTOC, TRIMDATA, [PAG], [PAGI],
                [FLXF1R], [FLXF1I], [FLXF2R], [FLXF2I], [MGG],
                [UFX1R], [UFX1I], [UFX2R], [UFX2I], BGPDT(BC),
                FLEXLOAD, SYMTRN(BC), ACSMTR(BC), NEWITER );
```

BCID	Boundary condition number (Integer, Input)
SUB	Subscript number of subcases considered in this call (Integer, Input)
TRIMTOC	Relation contains trim parameter table of contents. (Text, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Text, Input)
[PAG]	G-set aerodynamic load at real side. (Text, Input)
[PAGI]	G-set aerodynamic load at image side. (Text, Input)
[FLXF1R]	Real side G-set flexible increment force matrix which columns are related to trim configuration parameter . (Text, Input)
[FLXF1I]	Image side G-set flexible increment force matrix which columns are related to trim configuration parameter . (Text, Input)
[FLXF2R]	Real side G-set flexible increment force matrix which columns are related to rigid body acceleration . (Text, Input)
[FLXF2I]	Image side G-set flexible increment force matrix which columns are related to rigid body acceleration . (Text, Input)
[MGG]	G-set mass matrix. (Text, Input)
[UFX1R]	Real side F-set flexible deflection matrix which columns are related to trim configuration parameter . (Text, Input)
[UFX1I]	Image side F-set flexible deflection matrix which columns are related to trim configuration parameter. (Text, Input)
[UFX2R]	Real side F-set flexible deflection matrix which columns are related to rigid body acceleration. (Text, Input)
[UFX2I]	Image side F-set flexible deflection matrix which columns are related to rigid body acceleration. (Text, Input)

BGPDT(BC) — Relation of basic grid point coordinate data, where BC represents the MAPOL boundary condition loop index number (Input)

FLEXLOAD — Group contains flexible trim parameter load vector entities. (Text, Output)

SYMTRN(BC) — Logical denoting the presence of a "full" structural model instantiated from a centerline symmetric structural model via substructuring techniques (Logical, Input)

ACSMTR(BC) — Logical denoting the presence of a "full" structural model instantiated from a centerline symmetric structural model via substructuring techniques while QUADPAN method is used to create aerodynamic model. (Logical, Input)

NEWITER — Logical flag indicating the start of new design iteration (Logical, Input)

Application Calling Sequence:

None

Method:

This module begins by opening input matrices **PAG, FLXF1R, FLXF2R, UFX1R, UFX2R**. If it is substructure reflection case, matrices for the image side **PAGI, FLXF1I, FLXF2I, UFX1I, UFX2I** are also opened. Then, the module loops through trim configuration parameters and rigid body accelerations.

For each of these, the trim parameter or acceleration is obtained with its name and value, along with the aerodynamic model name, method name, Mach number, symmetric flag, pressure, and velocity. Then, **FLEXLOAD** group member matrices **FLXFRC** and **FLXDEF** are created. Relational entity **BGPDT** is loaded into group **FLEXLOAD** during the first pass.

If there is no substructure reflection, then group member matrix **FLXFRC** is a single column matrix for each trim parameter:

```
[FLXFRC] = [PAG] + [FLXF1R]
```

If there is a substructure reflection, then group member matrix **FLXFRC** is a two-column matrix for each trim parameter. The first column of matrix **FLXFRC** is:

```
= [PAG] + [FLXF1R]
```

and the second column is:

`= [PAGI] + [FLXF1I]`	if **QUADPAN** and Asymmetric trim
`= [PAG] + [FLXF1R]`	if **USSAERO** and Symmetric trim parameter
`= - [PAG] + [FLXF1I]`	if **USSAERO** and Asymmetric trim parameter

For each rigid body acceleration, **FLXFRC** is computed from multiplying g-size mass matrix **MGG** with g-size rigid body mode matrix **D**. Matrix **FLXFRC** is loaded into group **FLEXLOAD** for each

trim parameter and rigid boy acceleration. If there is no substructure reflection, then group member matrix **FLXDEF** is a single column matrix

```
[FLXDEF] = [UFX1R]        for trim parameter
[FLXDEF] = [UFX2R]        for rigid body acceleration
```

If there is a substructure reflection, then group member matrix **FLXDEF** is a two-column matrix. The first column is:

```
= [UFX1R]     for trim parameter
= [UFX2R]     for rigid body acceleration
```

and, the second column is:

```
= [UFX1I]     for trim parameter
= [UFX2I]     for rigid body acceleration
```

Matrix **FLXDEF** is loaded into group **FLEXLOAD** for each trim parameter and rigid body acceleration.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **FTRIMDRV**

Entry Point: **TRIMDR**

Purpose:

To examine and build groups for trim solver.

MAPOL Calling Sequence:

```
CALL FTRIMDRV ( BCID, SUB, TRIMDATA, METHOD, MODEL, MACH, SYM,
                SAMODEL, SAEMODEL, STDYGEOM, RIGDALOD, RIGDSLOD,
                FLEXLOAD, AICMAT, AEROGRID, CAEROBOX, SACOMPS, SAGEOM,
                [AIC], [AAIC], [ASAIC] );
```

BCID	Boundary condition number (Integer, Input)
SUB	Subscript number of subcases considered in this call (Integer, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Text, Input)
METHOD	The name of the method which created the aero model. (Text, Output)
MODEL	Name of the current model. (Text, Output)
MACH	Mach number. (Real, Output)
SYM	Trim symmetry flag. (Integer, Output) =-1 Antisymmtric = 0 Asymmetric = 1 Symmetric
SAMODEL	Group relation contains steady aerodynamic model group. (Text, Output)
SAEMODEL	Group relation contains steady aeroelastic model group. (Text, Output)
STDYGEOM	Group relation contains steady aerodynamic geometry entities. (Text, Output)
RIGDALOD	Group relation contains rigid aerodynamic trim parameter load. (Text, Output)
RIGDSLOD	Group relation contains rigid structural load. (Text, Output)
FLEXLOAD	Group contains flexible trim parameter load vector entities. (Text, Output)
AICMAT	Group relation contains AIC matrices. (Text, Output)
AEROGRID	Group member entity contains aerodynamic grid geometry relation. (Text, Output)
CAEROBOX	Group member entity contains aerodynamic connection relation. (Text, Output)

`SACOMPS`	Group member entity contains aerodynamic components relation. (Text, Output)
`SAGEOM`	Group member entity contains aerodynamic geometry relation. (Text, Output)
`[AIC]`	Group member entity contains the steady aerodynamic influence coefficient matrix for **SYM**metric Mach numbers. (Text, Output)
`[AAIC]`	Group member entity contains the steady aerodynamic influence coefficient matrix for **ANTISYM**metric Mach numbers. (Text, Output)
`[ASAIC]`	Group member entity contains the steady aerodynamic influence coefficient matrix for **ASYM**metric Mach numbers. (Text, Output)

<u>Application Calling Sequence:</u>

None

<u>Method:</u>

The module first gets model name, method name, Mach number and symmetry flag for the current subscript. Then, it generates **SAEMODEL** group name from model name. If this model is partially or fully generated by solution control model assembly, group API routines are called to create the group. Finally all trim solver related groups and member entities are created or obtained.

<u>Design Requirements:</u>

None

<u>Error Conditions:</u>

None

Engineering Application Module: **FTRIMOPT**

Entry Point: **FTRMOP**

Purpose:

To solve the trim equation for steady aeroelastic trim analyses while optimizing a user-defined objective function subject to user-defined constraints

MAPOL Calling Sequence:

```
CALL FTRIMOPT ( NITER, BCID, SUB, SYM, QDP, TRIMDATA, TRIMRSLT,
                TRIMTOC, [LHSA(BC,SUB)], [RHSA(BC,SUB)], [AAR],
                [DELTA(SUB)], [PRIGID], [R33], BMSTDATA );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **FLEXTRIM** subcases considered in this call (Integer, Input)
SYM	The symmetry flag for the current **FLEXTRIM** subcases (Integer, Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMRSLT	Relation containing **TRIMDATA** and results of the trim solutions (Output)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
[LHSA(BC,SUB)]	Matrix of modified inertia coefficients (Input), where **BC** represents the MAPOL boundary condition loop index number
[RHSA(BC,SUB)]	Matrix of applied load vectors reduced to the r-set (Input), where **BC** represents the MAPOL boundary condition loop index number
[AAR]	Matrix of acceleration vectors (Output)
[DELTA(SUB)]	Matrix of configuration parameters (Output)
[PRIGID]	Rigid Load Matrix (Input)
[R33]	Reduced rigid body mass matrix (Input)
BMSTDATA	Relation containing the trim parameter BMST component loads data (Input)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **FLEXTRIM** subcases in the current boundary condition. Then, the **TRIMTOC** relation is read into memory. The **TRIMDATA** and **TRIMRSLT** relations are read for all entries that have the current subscript value. Other trim data from **BMSTDATA, DRAGDATA, TFUNC, AEROS, CONEFFS,** and **CONLINK** are also read into memory.

Then an evaluation of the trim data is done to determine the number of trim subcases that will be solved during this pass (for the current subscript). The **SUPCHK** utility is used to evaluate the **SUPORT** conditions to ensure (again) that is satisfies the requirements of the **TRIM** solver and to get the names and DOFs of the supported degrees of freedom. Then, after creating needed scratch entities, the grand loop on the trim subcases begins.

Each trim subcase must be solved separately because of the options for control effectiveness and control linking. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** id of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **FLEXTRIM** entries in **CASE** for each **BC**). The trim optimization data for the current **TRIM** case is read into memory from the **TRIMOPT** and **TODVPRM** relations. If a **TRIMOPT** entry with the current **TRIM** id is not found, the trim case is skipped. Then, the **AROLNK** routine is called to assemble a linking matrix of control effectiveness factors and linking relationships for the current subscript such that:

{δ} = [TLINK] * DELRED

where the **DELRED** matrix is reduced to only the active trim parameters and the effectiveness factors have been included. Then the rigid and flexible loads are hit with the linking matrix to reduce the problem to the relevant configuration parameters:

P2RED = P2 * TLINK

RHSRED = RHS * TLINK

P2RED and **RHSRED** contain one row for each structural acceleration and one column for each non-acceleration label on the trim entry. This means that the total number of stability parameters (either fixed or free) is the number of columns in **P2** and **RHS**. Further, the order of the parameters is the order given on the **TRIM** tuples.

Now the trim equations can be assembled. From the input, we have the relationship

$$\begin{bmatrix} LHS_{ff} & LHS_{fk} \\ LHS_{kf} & LHS_{kk} \end{bmatrix} \begin{bmatrix} AR_{free} \\ AR_{known} \end{bmatrix} = \begin{bmatrix} RHS_{fu} & RHS_{fs} \\ RHS_{ku} & RHS_{ks} \end{bmatrix} \begin{bmatrix} DEL_{u} \\ DEL_{s} \end{bmatrix}$$

Where:	Represents:
F+K	Number of **SUPORT** point DOFs
F	Set of free accelerations, **AR**
K	Set of known **(FIXED)** accelerations, **AR**
U+S	Number of **AERO** parameters
U	Set of unknown parameters
S	Set of set **(FIXED)** parameters

The free accelerations and unknown parameters, or deltas, are the trim optimization design variables denoted on the **TODVPRM** bulk data entries associated with this **TRIM** case. These equations must be rearranged to get free accelerations and unknown deltas on the same side of the equation:

$$\begin{bmatrix} LHS_{ff} & -RHS_{fu} \\ LHS_{uf} & -RHS_{uu} \end{bmatrix} \begin{bmatrix} AR_{free} \\ DEL_u \end{bmatrix} = \begin{bmatrix} -LHS_{kk} & RHS_{ks} \\ -LHS_{sk} & RHS_{ss} \end{bmatrix} \begin{bmatrix} AR_k \\ DEL_s \end{bmatrix}$$

and we must handle the degenerate case where all accelerations or all deltas are known.

Following rearrangement of the equations, the unknowns are solved for in the **TRMOPS/D** routine. First the rigid masses and loads, **P2RED** and **MRR**, are used to obtain the rigid trim and then the flexible inputs, **RHSRED** and **LHS**, are used for the "real" solution.

The trim optimization objective function is specified on the **TRIMOPT** bulk data entry and may be a trim parameter, **BMST** component load, drag, or a linear function of these. The trim optimization constraints are specified on the **TCONTRM**, **TCONBMST**, and **TCONFUNC** bulk data entries. The imbalance in the supported DOFs are automatically included in the set of optimization constraints. Calls to **MDOT** invoke the mathematical programming algorithm which solves the trim optimization problem.

After the flexible results are unscrambled, the rigid body accelerations (either input on the **TRIM** entry or output from the solution of the above) are stored on the **AAR** matrix. The trim parameters are operated on by the **TLINK** matrix to recover the full vector from the reduced set and the results are stored on the **DELTA** matrix. Then the results for the rigid and flexible trim are printed.

The module repeats the entire process for all the subcases that are associated with the current **SUB**script and then terminates.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: `GPIMPORT`

Entry Point: `IMPORT`

Purpose:
To execute solution control aerodynamic model assembly command to import a model a group.

MAPOL Calling Sequence:

```
CALL GPIMPORT;
```

Application Calling Sequence:

```
CALL IMPORT
```

Method:
The module first brings all **IMPORT** entries into memory. Then, for each **IMPORT** entry, the new group name, group type, old group name and logical database name are obtained. The table of contents in the old group is copied into new group.

If the group type is **SAMODEL**, the corresponding **SAEMODEL** group is created and copied from **SAMODEL** group.

Design Requirements:
None

Error Conditions:
None

Engineering Application Module: GRPARCHV

Entry Point: GRPARC

Purpose:
To execute solution control aerodynamic model assembly command for archiving groups.

MAPOL Calling Sequence:
```
CALL GRPARCHV;
```

Application Calling Sequence:
```
CALL GRPARC
```

Method:
The module first brings all **ARCHIVE** entries into memory. Then, for each **ARCHIVE** entry, the new group name, group type, old group name and logical database name are obtained. If the group to be archived does not exist, all the member groups and member entities, as well as the group itself, are physically copied into new group at the specified database.

If the group type is **SAMODEL**, the corresponding **SAEMODEL** group is created and copied from the **SAMODEL** group.

Design Requirements:
None

Error Conditions:
None

Engineering Application Module: **LODSAGRP**

Entry Point: **AGENLD**

Purpose:

To load member groups and member entities into SAMODEL model group.

MAPOL Calling Sequence:

```
CALL LODSAGRP (SAMODEL, NEWMODEL, METHOD, MACH, SYM, STDYGEOM,
               RIGDALOD, AICMAT, AEROGRID, CAEROBOX, SACOMPS, SAGEOM,
               [AIC], [AAIC], [ASAIC], RIGDSLOD, FULAERO);
```

SAMODEL	Group relation contains steady aerodynamic model group. (Text, Input)
NEWMODEL	Logical flag indicating new model. (Logical, Input)
METHOD	The name of the method which created the aero model. (Text, Input)
MACH	Mach number. (Real, Input)
SYM	Trim symmetry flag. (Integer, Input) =-1 Antisymmtric = 0 Asymmetric = 1 Symmetric
STDYGEOM	Group relation contains steady aerodynamic geometry entities. (Text, Output)
RIGDALOD	Group relation contains rigid aerodynamic trim parameter load. (Text, Output)
AICMAT	Group relation contains AIC matrices. (Text, Output)
AEROGRID	Group member entity contains aerodynamic grid geometry relation. (Text, Output)
CAEROBOX	Group member entity contains aerodynamic connection relation. (Text, Output)
SACOMPS	Group member entity contains aerodynamic components relation. (Text, Output)
SAGEOM	Group member entity contains aerodynamic geometry relation. (Text, Output)
[AIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for SYMmetric Mach numbers. (Text, Output)
[AAIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for ANTISYMmetric Mach numbers. (Text, Output)
[ASAIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for ASYMmetric Mach numbers. (Text, Output)

`RIGDSLOD` Group relation contains rigid structural load. (Text, Output)

`FULAERO` Logical flag to indicate symmetry of geometry. (Logical, Output)

Application Calling Sequence:

None

Method:

This module first loads the **SAMODEL** group, if this subcase has new aerodynamic model. Then, all the related member groups, **STDYGEOM, RIGDALOD, RIGDSLOD** and **AIC** are also loaded into the **SAMODEL** group. Next, a corresponding **SAEMODEL** group is created and copied from the **SAMODEL** group. Finally, the member groups, **STDYGEOM**, and **AIC** are loaded with their member entities.

Design Requirements:

This module must be called after module **AROGNDRV**.

Error Conditions:

None

Engineering Application Module: **LODSPGRP**

Entry Point: **SPLNLD**

Purpose:

To examine solution control steady aeroelastic cases and create SPLINE group and member entities.

MAPOL Calling Sequence:

```
CALL LODSPGRP ( MINDEX, GOSPLINE, MODEL, SAEMODEL, SPLINE, FLXELOAD,
                [GTKG], [GSTKG], [GPTKG) );
```

MINDEX	Mach number index for the current pass. (Integer, Input)
GOSPLINE	Logical flag to generate spline matrices. (Logical, Input)
MODEL	Name of the current model. (Text, Input)
SAEMODEL	Group relation contains steady aeroelastic model group. (Text, Input)
SPLINE	Group relation contains spline matrices. (Text, Output)
FLEXLOAD	Group relation contains flexible trim parameter load vectors. (Text, Output)
[GTKG]	The matrix of splining coefficients relating the aeroelastic pressure increments to forces at the structural grids — the aeroelastic load increment spline (Input)
[GSTKG]	The matrix of splining coefficients relating the structural displacements to the streamwise slopes of the aerodynamic boxes — the aeroelastic slope increment spline (Input)
[GPTKG]	The matrix of splining coefficients relating the rigid aerodynamic pressures to forces at the structural grids — the aerodynamic load spline (Input)

Application Calling Sequence:

None

Method:

This module first loads the groups **SPLINE** and **FLEXLOAD** into the **SAEMODEL** group. Then, the member matrices **GTKG, GSTKG, GPTKG** are loaded into **SPLINE** group.

Design Requirements:

This module must be called first before calling **USSAERO/QUADPAN** and **LODSAGRP** modules.

Error Conditions:

None

Engineering Application Module: **MAKDFU**

Entry Point: **MAKDFU**

Purpose:

To assemble the sensitivities to the displacements of active stress and displacement constraints in the current active boundary condition.

MAPOL Calling Sequence:

```
CALL MAKDFU ( NITER, BC, GSIZEB, [SMAT], [NLSMAT], SMATCOL, NLSMTCOL,
              [GLBSIG], [NLGLBSIG], CONST, BGPDT, [DFDU], ACTUAGG,
              SUB, [GLBSIGI], [NLGBSIGI], [DFDUI], ACTUAGGI,
              SYMTRN(BC) );
```

NITER	Design iteration number (Integer, Input)
BC	Boundary condition identification number (Integer, Input)
GSIZEB	The size of the structural set (Integer, Input)
[SMAT]	Matrix entity containing the linear portion of the sensitivity of the stress and strain components to the global displacements (Input)
[NLSMAT]	Matrix entity containing the nonlinear portion of the sensitivity of the stress and strain components to the global displacements (Input)
SMATCOL	Relation containing matrix **SMAT** column information (Character, Input)
NLSMTCOL	Relation containing matrix **NLSMAT** column information (Character, Input)
[GLBSIG]	Matrix of stress/strain components for all the applied linearly designed stress constraints for the current boundary condition (Input)
[GLBSIGI]	Image side matrix of stress/strain components for all the applied linearly designed stress constraints for the current boundary condition for steady aerodynamic substructure reflection (Input)
[NLGLBSIG]	Matrix of stress/strain components for all the applied nonlinearly designed stress constraints for the current boundary (Input)
[NLGBSIGI]	Image side matrix of stress/strain components for all the applied nonlinearly designed stress constraints for the current boundary condition for steady aerodynamic substructure reflection (Input)
CONST	Relation of constraint values (Input)
BGPDT(BC)	Relation of basic grid point coordinates (Character,Input)
[DFDU]	Matrix containing the sensitivities of active displacement and/or stress-strain constraints to the displacements (Output)

`[DFDUI]`	Image side matrix containing the sensitivities of active displacement and/or stress-strain constraints to the displacements for steady aerodynamic substructure reflection (Output)
ACTUAGG	Logical flag to indicate whether any **DFDU** terms exist (Logical, Output)
ACTUAGGI	Logical flag to indicate whether any **DFDUI** terms exist for steady aerodynamic substructure reflection (Logical, Output)
SUB	An optional flag which indicates whether statics or static aeroelasticity is associated with the constraints in this call. The discipline flag `0` if STATICS `i` subscript identifier, SUB, of the aeroelastic subcases if SAERO (Integer, Input)
SYMTRN(BC)	Logical flag to indicate steady aerodynamic substructure reflection. (Logical, Output)

Application Calling Sequence:

None

Method:

For the current active boundary condition, the **MAKDFU** module begins by processing the active displacement constraints. The **CONST** relation is queried for all active displacement constraints **(CTYPE=3)**. Each tuple that qualifies the active condition is processed using the **PNUM** attribute to position to the appropriate location within the **DCENT** entity. The **DCENT** terms are loaded in the DFDU matrix in the order that active displacement constraints are encountered in the **CONST** relation. Constraints are evaluated for each load condition within the active boundary condition in constraint type order. The **DFDU** matrix is thus also formed in this order but the inactive constraints are ignored. After processing the active displacement constraints (if any), the **MAKDFU** module processes the active stress/strain constraints. The **CONST** relation is conditioned to retrieve the active stress and/or principal strain constraints **(CTYPE's 4, 5 and 6)**. For each active constraint, the current boundary condition number and the load condition number (stored on the **CONST** relation in the **SCEVAL** module) are used to determine the column number of the **SMAT** or **NLSMAT** matrix that holds the sensitivity of the current stress term to the displacements. Having recovered the **SMAT** or **NLSMAT** columns for the current active constraint, the **DFDU** terms are computed based on the element type and constraint type. Where the sensitivity is a function of the stress/strain values, the appropriate rows of the **GLBSIG** or **NLGLBSIG** column associated with the current boundary condition/load condition/discipline are retrieved for use in the computations.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **MKDFSV**

Entry Point: **MKDFSV**

Purpose:

To calculate matrix **DFSV** which contains the S-matrix derivatives related to active stress/strain constraints. The stress/strain constraints are functions of the product of the S-matrix and the displacements. The sensitivities of these constraints to the designed variables is decomposed into two parts. The first is a function of the product of the S-matrix derivatives and displacements, and the second is a function of the product of the S-matrix and the displacement derivatives.

MAPOL Calling Sequence:

```
CALL MKDFSV ( NITER, BC, GSIZEB, [NLGLBSIG], CONST, [NLSMAT],
              NLSMTCOL, [UGA], DESLINK, DSCFLG, NDV, GLBDES, LOCLVAR,
              [PTRANS], [DFSV], DELTA, [UGAI], [DFSVI] );
```

NITER	Design iteration number (Integer, Input)
BC	The MAPOL boundary condition loop index number (Integer, Input)
GSIZEB	Number of dofs in the structural set (Integer, Input)
[NLGLBSIG]	Stress vectors for design variable nonlinearly constrained elements (Character, Input)
CONST	Relation of constraints (Character, Input)
[NLSMAT]	Matrix entity containing the nonlinear portion of the sensitivity of the stress and strain components to the global displacements (Character, Input)
NLSMTCOL	Relation containing matrix **NLSMAT** column information (Character, Input)
[UGA]	Active displacement vectors at current boundary condition (Character, Input)
[UGAI]	Image side active displacement vectors at current boundary condition for steady aerodynamic substructure reflection (Character, Input)
DESLINK	Relation of design variable linking
DSCFLG	Discipline flag (Integer, Input) 0 statics ≠0 static aeroelasticity
NDV	Number of design variables (Integer, Input)
GLBDES	Global design variable relation (Character, Input)
LOCLVAR	Local design variable relation (Character, Input)
[PTRANS]	Design variable linking matrix (Character, Input)

`[DFSV]`	Matrix contains S-matrix derivatives related active stress/strain constraints (Character, Output)
`[DFSVI]`	Image side matrix contains S-matrix derivatives related active stress/strain constraints for steady aerodynamic substructure reflection (Character, Output)
`DELTA`	The relative design variable increment for finite difference computation. (Real, Input)

<u>Application Calling Sequence:</u>
None

<u>Method:</u>

This module first gets the **DVID** list from **GLBDES**, it then gets **EST** entries for nonlinearly designed constraint QUAD4 and TRIA3 elements and places them into lists in memory. Then the module determines the number of active displacement constraints, and gets active stress and strain constraints for this design iteration. Null columns are stored in **DFSV** corresponding to active displacement constraints so that **DFSV** will be compatible in module **MKAMAT**. For each active stress/strain constraint, the following operations are applied. If it is a linearly designed constraint, a null column is stored in **DFSV**, otherwise, matrix **DSDT** which contains the sensitivities of the nonlinear S-matrix to the related local design variables is computed for the element related to this constraint. Matrix DSDV which contains the sensitivities of the nonlinear S-matrix to the related global design variables is computed by using the **DSDT** matrix and design variable linking factors from **DESLINK**. Matrix **DSVU** is the multiplication of the transposed active displacement vector **UGA** times **DSDV**. The **DFSV** term at the row number corresponding to that active constraint is computed from **DSVU**, the constraint value, the related stress/strain values from **NLGLBSIG**, and the constraint allowables.

<u>Design Requirements:</u>
1. This module must be called prior to MKAMAT and after the active displacement vector is available.

<u>Error Conditions:</u>
None

Engineering Application Module: **OFPAEROM**

Entry Point: **OFPARO**

Purpose:

This module solves for the static aero applied loads on the aero boxes and for the displacements on the aero boxes to satisfy the **AIRDISP** and **TPRESSURE** print/punch requests. It loads the **OAGRDLOD** and **OAGRDDSP** relation.

MAPOL Calling Sequence:

```
CALL OFPAEROM ( NITER, BCID, MINDEX, SUB, GSIZE, GEOMSA, TRIMDATA,
                [GTKG], [GSTKG], QDP, [SAROLOAD], [DELTA(SUB)],
                [AIC], [UAG(BC)], OAGRDLOD, OAGRDDSP );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
MINDEX	Mach number index associated with the current subscript (Integer, Input)
SUB	Current Mach number subscript number (Integer, Input)
GSIZE	Number of g-set DOF's including any that may have been added by **GDR** (Integer, Input)
GEOMSA	A relation describing the aerodynamic boxes for the steady aerodynamics **MODEL**. The location of the box centroid, normal and pitch moment axis are given. It is used in splining the aerodynamics to the structure and to map responses back to the aerodynamic boxes (Input)
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
[GTKG]	The matrix of splining coefficients relating the aerodynamic pressures to forces at the structural grids (Input)
[GSTKG]	The matrix of splining coefficients relating the structural displacements to the streamwise slopes of the aerodynamic boxes (Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
[SAROLOAD]	Matrix containing the aerodynamic forces for unit configuration parameters for the current Mach number index. (Input)
[DELTA(SUB)]	Matrix containing the set of configuration parameters representing the user input fixed values and the trimmed unknown values for the SUB subscript's trim cases (Input)
[AIC]	Matrix containing the steady aerodynamic influence coefficients for either symmetric or antisymmetric Mach numbers as appropriate for the symmetry of the cases in the current boundary condition (Input)

[UAG(BC)]	Matrix of static displacements for all **SAERO** subcases in the current boundary condition in the order the subcases appear in the CASE] relation (Input), where **BC** represents the MAPOL boundary condition loop index number
OAGRDLOD	A relation containing the rigid, flexible correction and flexible forces and pressures for each **SAERO** subcase for the trimmed configuration parameters. Outputs are for the aerodynamic elements whose **TPRESSURE** output was requested in Solution Control. These constitute the loads of the "trimmed" state of the configuration. (Output)
OAGRDDSP	A relation containing the displacements for each **SAERO** subcase's set of configuration parameters for the aerodynamic elements whose **AIRDISP** output was requested in Solution Control. These constitute the trimmed displacements of the aerodynamic **MODEL**. (Output)

Application Calling Sequence:

None

Method:

The **CASE** relation is read to obtain the list of all **SAERO** subcases for the current boundary condition. The **AIRDISP** and **TPRESSURE** print/punch requests are checked and the module terminates if no output requests exist.

If output is needed, the **TRIM** relation is read to obtain the subscript values of each subcase. A partitioning vector is formed as the **TRIM** data are searched to extract the proper columns from the **UAG** matrix for the subcases associated with the current **SUB** value. Then, for each subcase to be processed, the particular print and punch requests are evaluated and, in the most general case, the following are computed:

Rigid Air Loads:

```
= QDP*[AIRFRC][DELTA]
```

Flexible Correction to the Rigid Air Loads:

```
= QDP*[AICMAT]T[GSTKG]T[UAG]
```

Total Applied Air Loads:

```
= Rigid + Flexible
```

Displacements on the aero boxes

```
= [GTKG]T[UAG]
```

where in each case the **DELTA** and **UAG** matrices are partitioned to include only the relevant subcases for the current subscript. Finally, the scratch matrices on which these results reside are read and output to the **OAGRDLOD** and **OAGRDDSP** relations for the loads and displacements, respectively.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **OFPALOAD**

Entry Point: **OFPALD**

Purpose:

Solves for the static aero applied loads and SPC forces to satisfy the print/punch requests. The resultant loads are written to the **OGRIDLOD** relation.

MAPOL Calling Sequence:

```
CALL OFPALOAD ( NITER, BCID, MINDEX, SUB, GSIZE, TRIMDATA, BGPDT(BC),
                [GPTKG], [GTKG], [GSTKG], QDP, [SAROLOAD],
                [DELTA(SUB)], [AIC], [UAG(BC)], [MGG], [AAG(BC)],
                [KFS], [KSS], [UAF], [YS(BC)], [PNSF(BC)], [PGMN(BC)],
                [PFJK], NGDR, USET(BC), OGRIDLOD );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
MINDEX	Mach number index for the current subscript value (Integer, Input)
SUB	Subscript number of SAERO subcases considered in this cal. (Integer, Input)
GSIZE	Number of degrees of freedom in the g-set including those that may have been added by GDR (Integer, Input)
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
BGPDT(BC)	Relation of basic grid point data for the boundary condition (including any extra points and GDR scalar points which may be added by the GDR module) (Input), where **BC** represents the MAPOL boundary condition loop index number
[GTKG]	The matrix of splining coefficients relating the aerodynamic pressures to forces at the structural grids (Input)
[GSTKG]	The matrix of splining coefficients relating the structural displacements to the streamwise slopes of the aerodynamic boxes (Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
[SAROLOAD]	Matrix containing the aerodynamic forces for unit configuration parameters for the current Mach number index. (Input)
[DELTA(SUB)]	Matrix containing the set of configuration parameters representing the user input fixed values and the trimmed unknown values for the SUB subscript's trim cases (Input)
[AIC]	Matrix containing the steady aerodynamic influence coefficients for either symmetric or antisymmetric Mach numbers as appropriate for the symmetry of the cases in the current boundary condition (Input)

[UAG(BC)]	Matrix of static displacements for all SAERO subcases in the current boundary condition in the order the subcases appear in the CASE relation (Input), where BC represents the MAPOL boundary condition loop index number
[MGG]	Mass matrix in the g-set (Input)
[AAG(BC)]	Matrix of accelerations for all SAERO subcases in the current boundary condition in the order the subcases appear in the **CASE** relation (Input), where BC represents the MAPOL boundary condition loop index number
[KFS]	The off-diagonal matrix partition of the independent degrees of freedom that results from the SPC partitioning (Input)
[KSS]	The dependent DOF diagonal matrix partition of the independent degrees of freedom that results from the SPC partitioning (Input)
[UAF]	Matrix of free (f-set) static displacements for all SAERO subcases in the current boundary condition in the order the subcases appear in the **CASE** relation (Input)
[YS(BC)]	Vector of enforced displacements for the boundary condition (one column) (Input)
[PNSF(BC)]	Partitioning vector to divide the independent DOFs into the free and SPC DOFs (Input), where BC represents the MAPOL boundary condition loop index number
[PGMN(BC)]	Partitioning vector to divide the g-set DOFs into the MPC and independent DOF's (Input), where BC represents the MAPOL boundary condition loop index number
[PFJK]	Partitioning vector to divide the f-set DOFs that may include GDR generated scalar points into the original f-set DOF's
NGDR	Denotes dynamic reduction in the boundary condition. (Input, Integer) =0 No GDR ≠1 GDR is used
USET(BC)	The unstructured entity of DOF masks for all the points in the current boundary conditions (Input), where BC represents the MAPOL boundary condition loop index number
OGRIDLOD	Relation of loads on structural grid points (Output)

Application Calling Sequence:

None

Method:

First the **CASE** relation entries for SAERO subcases in the current boundary condition are read. Then the TRIM relation is read to determine which subcases are associated with the current subscript value. Then the output **LOAD** and **SPCF** print/punch requests are examined to see if any further work is

needed. If no print or punch requests are needed for the subcases associated with the **SUB'th** subscript, control is returned to the MAPOL sequence.

If **SPCF** requests exist, the preliminary computations are performed in the ARSPCF module. It computes:

$$[QGV1] = [KFS]^T\{UF\} + [KSS]\{YS\}$$

for all the appropriate columns of UAF that are associated with the **SUB'th** subscript. The input YS vector is expanded to contain the correct number of columns.

Then the computation of the applied loads is done. First, the **BGPDT** data are read and the **OGRIDLOD** relation is opened for output. Then the loads for each subcase in the subscript is solved for subject to the existence of a print request for that subcase (either **LOAD** or **SPCF**). The following loads are computed:

Rigid Air Loads on the Structural Grids

$$= QDP*[GTKG][AIRFRC][DELTA]$$

Flexible Correction to the Rigid Air Loads

$$= QDP*[GTKG][AIC]^T[GSTKG]^T[UAG]$$

Total Applied Load

$$= Rigid + Flexible$$

Inertial Load

$$= -[MGG][AAG]$$

Where the appropriate inputs are not available, the computations are simply ignored with no warning. Thus, the optional calling arguments may be used to perform parts of the computations without requiring that all pieces be provided.

Then, the output **LOADs** matrices are opened and the **CASE LOADs** print and punch requests are used to load the **OGRIDLOD** relation with the **RIGID**, **FLEXIBLE**, **APPLIED** and **INERTIA** loads.

Finally, if any **SPCF** output requests exist, the **APPLIED** loads that were computed are combined with the **QGV1** terms to result in the **SPC** reaction forces:

$$[SPCF] = [QGV1] - [Applied\ load]$$

Design Requirements:

1. SPC force computations for other disciplines occur in the OFPSPCF module.

2. Only those arguments that are present will be used. If data are missing, the dependent terms will be omitted from the output.

Error Conditions:

None

Engineering Application Module: **OFPBMST**

Entry Point: **OFPBMS**

Purpose:

Calculate the trimmed BMST component load values, print out the results, and store the data on the **OBMSTLOD** relation

MAPOL Calling Sequence:

```
CALL OFPBMST ( NITER, BCID, SUB, QDP, [AAR], [DELTA(SUB)],
               TRIMDATA, TRIMTOC, BMSTDATA, TRMRIGD, OBMSTLOD );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **FLEXTRIM** subcases considered in this call (Integer, Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
[AAR]	Matrix of acceleration vectors (Output)
[DELTA(SUB)]	Matrix of configuration parameters (Output)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
BMSTDATA	Relation containing the trim parameter **BMST** component loads data (Input)
TRMRIGD	Logical indicating if the module is called within the **RIGDTRIM** discipline (Logical, Input)
OBMSTLOD	Relation containing the trimmed **BMST** component loads data (Output)

Application Calling Sequence:

None

Method:

The module first checks the **BMSTDATA** relation and exits if no **BMST** components have been defined. If **BMST** components have been defined, the **CASE** relation is read into memory for the current boundary condition and discipline, **RIGDTRIM** or **FLEXTRIM**, as indicated by the **TRMRIGD** logical. The **TRIMDATA** relation is read for all entries that have the current subscript value. The **BMST** and **TRIMTOC** relations are read into memory, as well.

Then an evaluation of the trim data is done to determine the number of trim subcases that were solved during this pass (for the current subscript). The **SUPCHK** utility is used to evaluate the **SUPORT**

conditions and to get the names and DOFs of the supported degrees of freedom. Then, the grand loop on the trim subcases begins.

Each trim subcase must be treated separately. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** id of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **TRIM** entries in **CASE** for each **BC**). Then, the trim parameter results from the `DELTA(SUB)` and `[AAR]` matrices associated with the current subcase are read into memory.

Now that all the parameter data for this **TRIM** case is available, the loop through each defined **BMST** component begins. The parameter **BMST** component loads data is read from the **BMSTDATA** relation and the BMST value is built up by parameter. The rigid body accelerations are retrieved from the **AAR** data and the non-acceleration parameters are retrieved from the `DELTA(SUB)` data. The rigid, rigid-splined, and applied trimmed BMST loads are calculated from the rigid, rigid-splined and flexible **BMSTDATA** parameter loads, respectively. The flexible increment trimmed BMST loads are calculated by subtracting the rigid-splined loads from the applied loads.

The trimmed **BMST** loads are added to the **OBMSTLOD** relation and printed out if a TRIM print has been requested. After each **BMST** component has been processed, the module exits.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **PARMBMST**

Entry Point: **PABMST**

Purpose:

Calculate the trim parameter BMST component loads data for use in trim optimization

MAPOL Calling Sequence:

```
CALL PARMBMST ( TRIMTOC, FLEXLOAD, [PAG], [PAGI] [SAROLOAD],
                STDYGEOM, SYMTRN(BC), ACSMTR(BC), BMSTDATA );
```

TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
FLEXLOAD	Name of the Flexible Trim Parameter Load Vector group (Input)
[PAG]	Matrix of rigid aerodynamic load at the structural g-set grid points for the real side (Input)
[PAGI]	Matrix of rigid aerodynamic load at the structural g-set grid points for the image side (Input)
[SAROLOAD]	Matrix of incremental non-dimensional rigid aerodynamic load at the aerodynamic panel control points (Input)
STDYGEOM	Name of the Steady Aerodynamic Model Geometry group (Input)
SYMTRN(BC)	Logical indicating if symmetric translation (substructuring) of a centerline symmetric structural model has occurred (Logical, Input)
ACSMTR(BC)	Logical flag indicating a QUADPAN-instantiated full aerodynamic model has been created from a user-defined half aerodynamic model for an asymmetric condition (Logical, Input)
BMSTDATA	Relation containing the trim parameter BMST component loads data (Output)

Application Calling Sequence:

None

Method:

The module first checks the **BMSTDATA** relation and exits if no BMST components have been defined. If BMST components have been defined, the **BMSTDATA** relation is read into memory, along with the **TRIMTOC** relation. The **AECOMPS** and **GEOMSA** entity names are retrieved from the **STDYGEOM** group, as well as the aerodynamic model type. The aerodynamic panel data is read from the **GEOMSA** relation and the **[SAROLOAD]** matrix is checked for compatibility. The **BGPDT** entity name is retrieved from the **FLEXLOAD** group and the basic grid point data is read from the **BGPDT** relation. The **[PAG]** and **[PAGI]** matrices are read into memory and checked for compatibility with the **BGPDT** data. The **BMSTDATA** relation is opened and prepared for loading.

Now the loop through each parameter in the **TRIMTOC** relation begins. The model/method/Mach and parameter information from **TRIMTOC** is loaded into the **BMSTDATA** projection space, along with the dynamic pressure and free stream velocity from the **FLEXLOAD** group. The next column

from the [SAROLOAD], [PAG] and [PAGI] matrices are read into memory and dimensionalized. The **FLXFRC** matrix names for this **TRIMTOC** entry are retrieved from the **FLEXLOAD** group. Matrices of multiple load types (**APPLIED, INERTIA,** and **STRUCTUR**) are summed together and dimensionalized. If substructuring occurred, both the real and image sides are processed.

Now that all the force data for this parameter is available, the loop through each defined BMST component begins. After the BMST label is loaded into the **BMSTDATA** projection space, the reference location for the component load is read from the **BGPDT** relation. If the BMST component load was requested in a non-basic coordinate system, the coordinate transformation matrix is read from the **CSTM** relation.

The rigid BMST parameter loads are calculated from the [SAROLOAD] matrix data using the panel list specified on the **BMST** entry. The rigid-splined and flexible loads are calculated from the [PAG], [PAGI], and **FLXFRC** matrices, respectively, using the grid list specified on the **BMST** entry. The **COMPLD** module is called to calculate the component loads. After each component load is calculated, the **BMSTDATA** relation is loaded with the data. This process continues for each **TRIMTOC** parameter and **BMST** definition.

Notes:

1. [PAGI] only exists if ACSMTR is true. This is the case of a user-defined half QUADPAN model that is reflected by QUADPAN to a full model for asymmetric conditions.

2. If SYMTRN(BC) is true, the **FLEXLOAD** group will contain double-column **FLXFRC** matrices for real and image sides, respectively.

3. For substructuring with a full aerodynamic model, real and image rigid loads will be generated. The first half of the [SAROLOAD] matrix will be used to create the real side and the second half will be used to create the image side.

4. For a half aerodynamic model, the [SAROLOAD] matrix will only contain loads on the real side of the aircraft. The rigid loads on the image side will be determined by reflection of load; equal loads for symmetric trim parameters, equal and opposite loads for antisymmetric trim parameters. Asymmetric trim parameters are not allowed for the case of substructuring with a half aerodynamic model.

5. Although only the flexible BMST parameter loads will be used for trim optimization, the rigid and rigid-splined BMST parameter loads will also be computed to allow the user to evaluate the spline on a component basis.

After all the BMST data has been calculated and loaded into the **BMSTDATA** relation, a summary of the parameter BMST component loads is printed. When this is complete, the module exits.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **PRETRM**

Entry Point: **PRETRM**

Purpose:

Perform trim preface checks

MAPOL Calling Sequence:

```
CALL PRETRM ( TRIMDATA, TRIMRSLT );
```

TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMRSLT	Relation containing **TRIMDATA** updated with trim optimization initial values, if applicable (Output)

Application Calling Sequence:

None

Method:

The module begins by bringing the **TRIMDATA** relation into memory and using it to initialize the **TRIMRSLT** relation which is also brought into memory. Then, the **CONVERT** relation is searched for a mass conversion factor. If one is found, it is used to convert the linear accelerations in the **TODVPRM** and **TCONTRM** relations to proper units. The **TRIMDATA** relation is examined to determine the total number of trim subcases and then the grand loop on trim subcases begins.

The processing of each trim case begins by aligning the **TRIMDATA** and **TRIMRSLT** data sets to the same trim case. After this is accomplished, the trim case is searched for any scheduled parameters as indicated by the character string 'SCHD' in the 'FREEI' field. The schedule data is retrieved from the **SCHEDULE** and **AEFACT** relations and checked for the proper number of schedule table values and valid functional parameter names.

The **TRIMOPT** relation is checked for an entry with the same trim identification. If one exists, the corresponding **TODVPRM** entries are searched for matching trim parameters. If a parameter match is found and an initial value is indicated on the **TODVPRM** entry, the **TRIMRSLT** relation is updated. If no parameter match is found, the **TODVPRM** parameter is added to the **TRIMRSLT** relation. User information messages are printed if inconsistencies between the **TRIMDATA** and **TODVPRM** parameter values are found.

(Note: The **TODVPRM** entry can specify either an initial parameter value or the character string 'TRIM'. If the character string 'TRIM' is specified, the result of the previous standard trim is used as the initial value for the trim optimization. The updated **TRIMRSLT** relation will be used to provide initial trim parameter values to the trim optimization module, **FTRIMOPT**.)

The module will terminate the program if fatal errors are found, otherwise the module exits and processing continues.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **QUADPAN**

Entry Point: **QUDPAN**

Purpose:

Process the Quadrilateral Panel Method steady aerodynamics (preface to static aeroelasticity discipline)

MAPOL Calling Sequence:

```
CALL QUADPAN ( MODEL, CASEID, CASE, MACH, AICSYM, AEROGRID,
               CAEROBOX, SACOMPS, SAGEOM, REFPARAM, [AIC], [AAIC],
               {ASAIC], [AIRFRC], RIGDALOD, FULAERO );
```

MODEL	Name of the current model. (Text, Input)
CASEID	Subcase identification number. (Integer, Input)
CASE	Relation containing the case control information. (Input)
MACH	Mach number. (Real, Input)
AICSYM	AIC matrix symmetry option. (Integer, Input) = - 2 None = - 1 Antisymmtric = 0 Asymmetric = 1 Symmetric = 2 Both symmetric and antisymmetric
AEROGRID	Group member entity contains aerodynamic grid geometry relation. (Text, Output)
CAEROBOX	Group member entity contains aerodynamic connection relation. (Text, Output)
SACOMPS	Group member entity contains aerodynamic components relation. (Text, Output)
SAGEOM	Group member entity contains aerodynamic geometry relation. (Text, Output)
REFPARAM	Group member relation entity contains nominal trim parameter settings. (Text, Output)
[AIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for SYMmetric Mach numbers. (Text, Output)
[AAIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for ANTISYMmetric Mach numbers. (Text, Output)
[ASAIC]	Group member entity contains the steady aerodynamic influence coefficient matrix for ASYMmetric Mach numbers. (Text, Output)

[AIRFRC]	Single-column matrix entity containing rigid aerodynamic loads for a trim parameter. (Text, Output)
RIGDALOD	Group relation contains rigid aerodynamic trim parameter loads. (Text, Output)
FULAERO	Logical indicating a full aerodynamic model. (Logical, Output)

Application Calling Sequence:

None

Method:

The IFP preface module reads the QUADPAN PACKET and stores the information for later QUADPAN operations. QUADPAN is called through the ASTROS MAPOL for each Mach number specified in an SAERO discipline.

The QUADPAN module first reads the input to allocate memory. Then QUADPAN processes the aerodynamic geometry and stores it in the **STDYGEOM** group. A summary of the geometry is printed to the standard ASTROS output. Geometry is checked for compliance between QUADPAN macro-elements known as PANELS. If tolerance checks reveal compliance failures, the run is terminated. Compliance (abutment) check summaries are printed to the standard ASTROS output. Once compliance checks are satisfied, QUADPAN processes rigid forces and stores them in the **AIRFRC** matrices of the **RIGDALOD** group along with associated parameters of onset flow conditions and control surface data. QUADPAN performs an outside loop through onset flow conditions (i.e. flow onset angles and rates) and an inside loop on control surface parameters. Flow condition summaries are printed to standard ASTROS output. After computing all of the flow conditions, QUADPAN computes the user specified aerodynamic influence coefficient matrices and stores the terms in the **AIC**, **AAIC** and/or **ASAIC** entities (depending on the symmetry options) of the **AIC** group. QUADPAN then exits.

Detailed information on the QUADPAN module is contained in References 3 and 4.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **RBMGEN**

Entry Point: **RBMGEN**

Purpose:
To generate G size rigid body mode matrix

MAPOL Calling Sequence:

```
CALL RBMGEN ( BGPDT(BC), GID, [RGBDMG] );
```

BGPDT(BC)	Relation contains basic grid point data table. (Text, Input)
GID	Identification number of the grid with respect to which the rigid body modes are taken. (Integer, Input)
[RGBDMG]	G size rigid body mode matrix. (Text, Output)

Application Calling Sequence:
None

Method:
The module begins by bringing **BGPDT** entries into memory. Then, it searches input reference grid pint in **BGPDT** entries and obtains its basic coordinates. For each point in **BGPDT**, the basic to global transformation matrix and the rigid body connection matrix are computed to generate the g size rigid body mode matrix.

Design Requirements:
None

Error Conditions:
None

Engineering Application Module: **RIGDSTAB**

Entry Point: **RGDSTB**

Purpose:
Generate rigid steady aerodynamic stability derivative coefficients

MAPOL Calling Sequence:

```
CALL RIGDSTAB ( BCID, SUB, TRIMTOC, [SAROLOAD], STDYGEOM, BGPDT(BC),
                QDP, STABCFA );
```

BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **FLEXTRIM** subcases considered in this call (Integer, Input)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
[SAROLOAD]	Matrix containing the incremental non-dimensional aerodynamic load at the aerodynamic panel control points for the current Model / Method / Mach combination (Force/QDP, from ARFMRG) (Input)
STDYGEOM	Name of the steady aerodynamic model geometry group (Input)
BGPDT(BC)	Relation of basic grid point coordinate data (Input), where **BC** represents the MAPOL boundary condition loop index number
QDP	Dynamic pressure associated with the current subscript (Real, Input)
STABCFA	Relation containing the rigid steady aerodynamic stability derivative coefficients (Output)

Application Calling Sequence:
None

Method:
The module begins by bringing the **TRIMTOC** data into memory. The reference grid point identification for the moment derivatives is retrieved from the **AEROS** relation and the coordinates of the reference location are retrieved from the **BGPDT** relation. The aerodynamic model symmetry, **MODELTYP**, is retrieved from the **STDYGEOM** group table of contents and used to determine if a full aerodynamic model is present either by definition or by reflection of a centerline symmetric model (**MODELTYP = 'ASYM'**).

The name of the **GEOMSA** relation containing the aerodynamic panel geometric defining data is retrieved from the **STDYGEOM** group and the **GEOMSA** data is read into memory. The **SAROLOAD** matrix is opened and checked for compatibility with the **GEOMSA** data.

The **TRIMTOC** data is searched for the base parameter as indicated by the character string **'BASE'** in the **PARMTYP** field. The base parameter name and value are loaded into the **STABCFA** projection space along with **QDP** and **SCFTYPE='RIGID'**.

Each entry in the **TRIMTOC** relation is then processed. The parameter names, values, symmetry, model, method, and Mach are loaded into the **STABCFA** projection space. The normalization factors are determined using the **AEROS** and **MODELTYP** information, with special treatment given to rate parameters. The **SAROLOAD** column corresponding to the current **TRIMTOC** entry is read into memory and the forces and moments are calculated and normalized. If the load vector is for a half model and its symmetry is symmetric or antisymmetric, the opposite symmetry coefficients are zeroed out. The final stability coefficient results are loaded into the projection space and added to the **STABCFA** relation.

After the last **TRIMTOC** entry has been processed, the module terminates.

Notes: 1) This module is only applicable within the flexible trim, FTRIM, discipline.
2) The control surface effectiveness factors are not included in the coefficient data.

Design Requirements:
None

Error Conditions:
None

Engineering Application Module: **RIGDTRIM**

Entry Point: **RGDTRM**

Purpose:

To solve the trim equation for rigid steady aerodynamic trim analyses.

MAPOL Calling Sequence:

```
CALL RIGDTRIM ( NITER, BCID, SUB, SYM, QDP, TRIMDATA, TRIMRSLT,
                TRIMTOC, [R33], [PRIGID], [AAR], [DELTA(SUB)] );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **RIGDTRIM** subcases considered in this call (Integer, Input)
SYM	The symmetry flag for the current **RIGDTRIM** subcases (Integer, Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMRSLT	Relation containing **TRIMDATA** and results of the trim solutions (Output)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
[R33]	Reduced rigid body mass matrix (Input)
[PRIGID]	Rigid Load Matrix (Input)
[AAR]	Matrix of acceleration vectors (Output)
[DELTA(SUB)]	Matrix of configuration parameters (Output)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **RIGDTRIM** subcases in the current boundary condition. Then, the **TRIMTOC** relation is read into memory. The **TRIMDATA** and **TRIMRSLT** relations are read for all entries that have the current subscript value. Other trim data from **AEROS**, **CONEFFS**, and **CONLINK** are also read into memory.

Then an evaluation of the trim data is done to determine the number of trim subcases that will be solved during this pass (for the current subscript). Then, after creating needed scratch entities, the grand loop on the trim subcases begins.

Each trim subcase must be solved separately because of the options for control effectiveness and control linking. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** id of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **RIGDTRIM** entries in **CASE** for each **BC**). The **TRIMR** entry is examined to determine the trim type requested and the degrees of freedom to use for the trim. Then, the **AROLNK** routine is called to assemble a linking matrix of control effectiveness factors and linking relationships for the current subscript such that:

$$\{\delta\} = [\mathbf{TLINK}] * \mathbf{DELRED}$$

where the **DELRED** matrix is reduced to only the active trim parameters and the effectiveness factors have been included. Then the rigid loads are hit with the linking matrix to reduce the problem to the relevant configuration parameters:

$$\mathbf{RHSRED} = \mathbf{PRIGID} * \mathbf{TLINK}$$

RHSRED contain one row for each structural acceleration and one column for each non-acceleration label on the trim entry. This means that the total number of stability parameters (either fixed or free) is the number of columns in **PRIGID**. Further, the order of the parameters is the order given on the **TRIM** tuples.

Now the trim equations can be assembled. From the input, we have the relationship

$$\begin{bmatrix} LHS_{ff} & LHS_{fk} \\ LHS_{kf} & LHS_{kk} \end{bmatrix} \begin{bmatrix} AR_{free} \\ AR_{known} \end{bmatrix} = \begin{bmatrix} RHS_{fu} & RHS_{fs} \\ RHS_{ku} & RHS_{ks} \end{bmatrix} \begin{bmatrix} DEL_u \\ DEL_s \end{bmatrix}$$

Where:	Represents:
F+K	Number of free DOFs
F	Set of free accelerations, **AR**
K	Set of known **(FIXED)** accelerations, **AR**
U+S	Number of **AERO** parameters
U	Set of unknown parameters
S	Set of set **(FIXED)** parameters

These equations must be rearranged to get free accelerations and unknown deltas on the same side of the equation:

$$\begin{bmatrix} LHS_{ff} & -RHS_{fu} \\ LHS_{uf} & -RHS_{uu} \end{bmatrix} \begin{bmatrix} AR_{free} \\ DEL_u \end{bmatrix} = \begin{bmatrix} -LHS_{kk} & RHS_{ks} \\ -LHS_{sk} & RHS_{ss} \end{bmatrix} \begin{bmatrix} AR_k \\ DEL_s \end{bmatrix}$$

and we must handle the degenerate case where all accelerations or all deltas are known.

Following rearrangement of the equations, the unknowns are solved for in the **ARTRMS/D** routine. The rigid masses and loads, **RHSRED** and **R33**, are used to obtain the rigid trim solution.

The trim results are unscrambled and the rigid body accelerations (either input on the **TRIM** entry or output from the solution of the above) are stored on the **AAR** matrix and the same is done with the trim parameters after the **TLINK** matrix is used to recover the full vector from the reduced set. Then the results for the rigid trim are printed.

The module repeats the entire process for all the subcases that are associated with the current **SUB**script and then terminates.

Design Requirements:
None

Error Conditions:
None

THIS MODULE IS NOT IMPLEMENTED.

Engineering Application Module: **RTRIMOPT**

Entry Point: **RTRMOP**

Purpose:

To solve the trim equation for rigid steady aerodynamic trim analyses while optimizing a user-defined objective function subject to user-defined constraints

MAPOL Calling Sequence:

```
CALL RTRIMOPT ( NITER, BCID, SUB, SYM, QDP, TRIMDATA, TRIMRSLT,
                TRIMTOC, [PRIGID], [R33], [AAR], [DELTA(SUB)],
                BMSTDATA );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **RIGDTRIM** subcases considered in this call (Integer, Input)
SYM	The symmetry flag for the current **RIGDTRIM** subcases (Integer, Input)
QDP	Dynamic pressure associated with the current subscript (Real, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMRSLT	Relation containing **TRIMDATA** and results of the trim solutions (Output)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
[PRIGID]	Rigid Load Matrix (Input)
[R33]	Reduced rigid body mass matrix (Input)
[AAR]	Matrix of acceleration vectors (Output)
[DELTA(SUB)]	Matrix of configuration parameters (Output)
BMSTDATA	Relation containing the trim parameter BMST component loads data (Input)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **RIGDTRIM** subcases in the current boundary condition. Then, the **TRIMTOC** relation is read into memory. The **TRIMDATA** and **TRIMRSLT** relations are read for all entries that have the current subscript value.

Other trim data from **BMSTDATA, DRAGDATA, TFUNC, AEROS, CONEFFS,** and **CONLINK** are also read into memory.

Then an evaluation of the trim data is done to determine the number of trim subcases that will be solved during this pass (for the current subscript). Then, after creating needed scratch entities, the grand loop on the trim subcases begins.

Each trim subcase must be solved separately because of the options for control effectiveness and control linking. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** id of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **FLEXTRIM** entries in **CASE** for each **BC**).). The **TRIMR** entry is examined to determine the trim type requested and the degrees of freedom to use for the trim. The trim optimization data for the current **TRIM** case is read into memory from the **TRIMOPT** and **TODVPRM** relations. If a **TRIMOPT** entry with the current **TRIM** id is not found, the trim case is skipped. Then, the **AROLNK** routine is called to assemble a linking matrix of control effectiveness factors and linking relationships for the current subscript such that:

$$\{\delta\} = [\mathbf{TLINK}] * \mathbf{DELRED}$$

where the **DELRED** matrix is reduced to only the active trim parameters and the effectiveness factors have been included. Then the rigid loads are hit with the linking matrix to reduce the problem to the relevant configuration parameters:

$$\mathbf{P2RED} = \mathbf{PRIGID} * \mathbf{TLINK}$$

RHSRED contain one row for each structural acceleration and one column for each non-acceleration label on the trim entry. This means that the total number of stability parameters (either fixed or free) is the number of columns in **PRIGID**. Further, the order of the parameters is the order given on the **TRIM** tuples.

Now the trim equations can be assembled. From the input, we have the relationship

$$\begin{bmatrix} LHS_{ff} & LHS_{fk} \\ LHS_{kf} & LHS_{kk} \end{bmatrix} \begin{bmatrix} AR_{free} \\ AR_{known} \end{bmatrix} = \begin{bmatrix} RHS_{fu} & RHS_{fs} \\ RHS_{ku} & RHS_{ks} \end{bmatrix} \begin{bmatrix} DEL_u \\ DEL_s \end{bmatrix}$$

Where:	Represents:
F+K	Number of **SUPORT** point DOFs
F	Set of free accelerations, **AR**
K	Set of known **(FIXED)** accelerations, **AR**
U+S	Number of **AERO** parameters
U	Set of unknown parameters
S	Set of set **(FIXED)** parameters

The free accelerations and unknown parameters, or deltas, are the trim optimization design variables denoted on the **TODVPRM** bulk data entries associated with this **TRIM** case. These equations must be rearranged to get free accelerations and unknown deltas on the same side of the equation:

$$\begin{bmatrix} LHS_{ff} & -RHS_{fu} \\ LHS_{uf} & -RHS_{uu} \end{bmatrix} \begin{bmatrix} AR_{free} \\ DEL_u \end{bmatrix} = \begin{bmatrix} -LHS_{kk} & RHS_{ks} \\ -LHS_{sk} & RHS_{ss} \end{bmatrix} \begin{bmatrix} AR_k \\ DEL_s \end{bmatrix}$$

and we must handle the degenerate case where all accelerations or all deltas are known.

Following rearrangement of the equations, the unknowns are solved for in the **TRMOPS/D** routine. The rigid masses and loads, **P2RED** and **R33**, are used to obtain the rigid trim solution.

The trim optimization objective function is specified on the **TRIMOPT** bulk data entry and may be a trim parameter, **BMST** component load, drag, or a linear function of these. The trim optimization constraints are specified on the **TCONTRM**, **TCONBMST**, and **TCONFUNC** bulk data entries. The imbalance in the supported DOFs are automatically included in the set of optimization constraints. Calls to **MDOT** invoke the mathematical programming algorithm which solves the trim optimization problem.

After the results are unscrambled, the rigid body accelerations (either input on the **TRIM** entry or output from the solution of the above) are stored on the **AAR** matrix. The trim parameters are operated on by the **TLINK** matrix to recover the full vector from the reduced set and the results are stored on the **DELTA** matrix. Then the results for the rigid trim are printed.

The module repeats the entire process for all the subcases that are associated with the current **SUB**script and then terminates.

Design Requirements:
None

Error Conditions:
None

THIS MODULE IS NOT IMPLEMENTED.

Engineering Application Module: **SAERODRV**

Entry Point: **SARODR**

Purpose:

MAPOL director for steady aeroelastic analyses.

MAPOL Calling Sequence:

```
CALL SAERODRV (BCID, SUB, LOOP, MINDEX, SYM, MACH, QDP, TRIMDATA,
               TRIMRSLT, METHOD, PRINT );
```

BCID	User defined boundary condition identification number (Integer, Input)
SUB	Current Mach number subscript number (Integer, Input)
LOOP	Logical flag indicating whether another subscript is required to complete the set of all subcases (Logical, Output)
MINDEX	Mach number index associated with the current subscript (Integer, Output)
SYM	**SYM**metry flag for the current subscript. (Integer, Output) 1 Symmetric ≠1 Antisymmetric
MACH	Mach number associated with the current subscript (Real, Output)
QDP	Dynamic pressure associated with the current subscript (Real, Output)
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
TRIMRSLT	Relation contains the **TRIM** results. (Input)
METHOD	The name of method which creates aerodynamic model. (Output)
PRINT	Optional print flag indicating that the summary of trim cases associated with the current pass (subscript) is to be printed to the standard output. (In the standard sequence, **PRINT** is used only during analysis not during sensitivity analysis) (Optional, Integer, Input)

Application Calling Sequence:

None

Method:

First the **CASE** relation is read to determine the **TRIM** ids and **SYM**metries of all SAERO cases in the current boundary condition. If any exist, the **TRIM** relation is opened and read into memory. Each trim entry referenced in **CASE** is then compressed into a format containing the **TRIM** id, Mach number, dynamic pressure, trim type, Mach number index, subscript and subcase id.

Once these data are collected, the **CASE** tuples read into memory are looped over to choose which **TRIM** cases are to be analyzed for this subscript value. There are four steps in choosing the proper trim cases:

1. Take the first SAERO subcase in CASE that has not been done on an earlier pass - cases already analyzed will reference trims with a "subscript" value that is not "null" (uninitialized) and that is less than the current value of **SUB** - on the first design iteration all subscript values will be "null"

2. Once the parent case is known, choose that case and all others with the same Mach, QDP and **TRMTYP**

3. Update the "subcript" attribute in **TRIM** to mark all the cases that are being processed. Also load the **SUBID** to assist in re-merging the answers into **CASE** subcase order

4. Check if any more saero subcases need to be processed and set the "loop" flag

After these steps have been completed, if the **PRINT** flag is nonzero, a summary of the selected **TRIMs** is printed to the output file.

Design Requirements:

1. The TRIM relation is assumed to contain NULL values for SUBSCRPT on the first subscript of the first design iteration (for OPTIMIZE boundary conditions) and for the first subscript of all ANALYZE boundary conditions.

Error Conditions:

None

Engineering Application Module: **SAEROMRG**

Entry Point: **SAROMR**

Purpose:

Merges the static aero results for each subscript (stored in the matrix **MATSUB**) into the **MATOUT** matrix in case order rather than subscript order for the **BCID**'th boundary condition.

MAPOL Calling Sequence:

```
CALL SAEROMRG ( BCID, SUB, TRIMDATA, [MATOUT], [MATSUB] );
```

`BCID`	User defined boundary condition identification number (Integer, Input)
`SUB`	Current Mach number subscript (Input, Integer)
`TRIMDATA`	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)
`[MATOUT]`	Merged output matrix reordered to be in **CASE** order for the current boundary condition (Input and Output)
`[MATSUB]`	Generic input matrix containing data for the current subscript value in **TRIM** id order of **TRIM** cases associated with the current subscript (Input)

Application Calling Sequence:

None

Method:

First the **CASE** relation is read to retrieve the **TRIM** id's for the SAERO subcases in the current boundary condition. Then the **TRIM** relation is read to obtain the subcase numbers associated with each **TRIM** id having the current **SUB**script value.

Then the **MATSUB** and **MATOUT** matrices are opened. If **MATOUT** is uninitialized, or if **SUB = 1**, it is initialized (flushed and the number of rows, precision and form set to those of **MATSUB**). If **MATOUT** already exists and has data in it, a scratch matrix is created to hold the final merged data.

For each SAERO **CASE** entry for the current boundary, the **TRIM** data are searched to determine the subscript number associated with the subcase. If the subscript is less than **SUB**, a column from **MATOUT** will be taken (it was stored there on an earlier pass). If the subscript is equal to SUB, it will be stored on the output matrix from **MATSUB**. If greater than **SUB**, it is ignored until later passes.

Once a column is identified as active in **MATSUB** (**PGAA** indicates active and subscript = **SUB**), an additional check is made to see if the column is active in **PGUA**. Only those columns that are active in **PGUA** are copied to **MATOUT**. This filtering is done to limit the amount of computational effort in the stress, strain and displacement constraint sensitivity computations that proceed using the **MATOUT** matrix. The **MATSUB** columns that are active due to **DCONTRM** constraints are no longer needed as these sensitivities are assumed to have been computed already in the AEROSENS module.

Once the final matrix is formed, if **MATOUT** had had data in it, the name of the scratch matrix that was loaded is switched with that of **MATOUT**. The scratch entity is then destroyed.

Design Requirements:

1. The assumption is that each MATSUB matrix contains the results from the "SUB"th subscript value in the order the trim id's for that SUB appear in the TRIM relation.

2. The same MATOUT matrix must be passed into the AROSNSMR module on each call since the columns associated with earlier subscript values are read from MATOUT into a scratch entity. The merged matrix that results then replaces the input MATOUT.

3. The AEROSENS module is called upstream of the AROSNSMR module to process active DCONTRM constraints for the current subscript. Thus, those columns that are active only for DCONTRM constraints may be filtered out for the downstream processing of stress, strain and displacement constraints.

Error Conditions:

None

Engineering Application Module: **SCEVAL**

Entry Point: **SCEVAL**

Purpose:

To compute the stress and/or strain constraint values for the statics or steady aeroelastic trim analyses in the current boundary condition.

MAPOL Calling Sequence:

```
CALL SCEVAL ( NITER, BCID, [UG(BC)], [SMAT], [NLSMAT], SMATCOL,
              NLSMTCOL, TREF, TREFD, [GLBSIG],[NLGLBSIG], CONST,
              DSCFLG, TRIMDATA, [UAGI(BC)], [GLBSIGI], [NLGBSIGI] );
```

NITER	Design iteration number (Integer, Input)
BCID	User defined boundary condition identification number (Integer, Input)
[UG(BC)]	The matrix of global displacements for all static applied loads in the current boundary condition (Input), where BC represents the MAPOL boundary condition loop index number.
[SMAT]	Matrix entity containing the linear portion of the sensitivity of the stress and strain components to the global displacements (Input)
[NLSMAT]	Matrix entity containing the nonlinear portion of the sensitivity of the stress and strain components to the global displacements (Input)
SMATCOL	Relation containing matrix **SMAT** column information (Character, Input)
NLSMTCOL	Relation containing matrix **NLSMAT** column information (Character, Input)
TREF	Unstructured entity containing the linearly designed element reference temperatures (Input)
TREFD	Unstructured entity containing the nonlinearly designed variable element reference temperatures (Input)
[GLBSIG]	Matrix of stress/strain components for all the applied linearly designed stress constraints for the current boundary condition (Output)
[NLGLBSIG]	Matrix of stress/strain components for all the applied nonlinearly designed stress constraints for the current boundary condition (Output)
CONST	Relation of constraint values (Const)
DSCFLG	The discipline flag (Integer, Input) 0 Statics >0 Static aeroelasticity
TRIMDATA	Relation contains the **TRIM** bulk data and related boundary condition, subcase and subscript information. (Input)

[UAGI(BC)]	The image side matrix of global displacements for all static applied loads in the current boundary condition for steady aerodynamic substructure reflection. (Input), where **BC** represents the MAPOL boundary condition loop index number.
[GLBSIGI]	Image side matrix of stress/strain components for all the applied linearly designed stress constraints for the current boundary condition condition for steady aerodynamic substructure reflection. (Output)
[NLGBSIGI]	Image side matrix of stress/strain components for all the applied nonlinearly designed stress constraints for the current boundary condition condition for steady aerodynamic substructure reflection. (Output)

Application Calling Sequence:

None

Method:

The SCEVAL module begins by determining if there are any stress constraints applied and any user functions which require element response functions. If any are found, execution continues.

First the **CASE** relation is read. Then, if the call is associated with SAERO disciplines, the **TRIM** relation is read to associate, for each subcase, the subcase id and the subscript id. Then an in-core table is formed that contains, for the subcases in this boundary condition the **DISFLAG**, **SUBSCRIPT**, and **THERMID**. The latter is for thermal load corrections to the stresses and strains. If any thermal load cases were found, the **GRIDTEMP** and **TREF** entities are opened.

If the current boundary condition is the first with stress or strain constraints, the running constraint type count variables are reinitialized for the current design iteration. This type count provides a link between the **ACTCON** print of design constraints and the debug print option supported by the SCEVAL module. If any thermal loads exist for the current boundary condition, the **GRIDTEMP**, **TREF** and **TREFD** entities are brought into memory to be available for the computation of the stress-free thermal strain correction to the element stresses. Once these preparations have been made, the **SMAT** and **NLSMAT** matrices of stress/strain sensitivities and the **GLBSIG** and **NLGLBSIG** matrices are opened and the **GLBSIG** and **NLGLBSIG** matrices are positioned to the proper columns to pack additional stress/strain components. Note that the **GLBSIG** and **NLGLBSIG** matrices store all the columns associated with the current boundary condition since they are required for the constraint sensitivity computations.

Finally, the **UG** matrix of global displacements is opened. For each column in the UG matrix, the matrix products

$$[GMA] = [SMAT]\{UG\} \text{ and } [NLGMA] = [NLSMAT]\{UG\}$$

are calculated to obtain the component stress or strain values for linearly designed elements and nonlinearly designed elements, respectively. Having calculated and stored in core these values, the element dependent constraint evaluation routines are called to process each constraint. Note that the order in which the element routines are called must be the same as the order the **SMAT** and **NLSMAT** columns were formed. That order is:

1. Bar elements, **BARSC** (Using both **GMA** and **NLGMA**)
2. Isoparametric quadrilateral membrane elements, **QD1SC** (Using **GMA** only)

3. Quadrilateral bending plate elements, **QD4SC** (Using both **GMA** and **NLGMA)**
4. Rod elements, **RODSC** (Using **GMA** only)
5. Shear panels, **SHRSC** (Using **GMA** only)
6. Triangular bending plate elements, **TR3SC** (Using both **GMA** and **NLGMA)**
7. Triangular membrane elements, **TRMSC** (Using **GMA** only)

On the first pass through the element dependent routines, all the xxxxEST tuples (i.e., **RODEST** and **TRMEMEST)** with nonzero stress/strain constraint flags are retrieved from the data base. For subsequent passes, this information is used directly from core. Each constraint is evaluated in turn with the stress components modified by the thermal stress correction if the displacement field includes thermal strain effects. The **CONST** relation is loaded with one tuple for each constraint as they are processed. When all the constraints have been evaluated for the current loading condition, the adjusted linear design variable and nonlinear design variable stress/strain constraint terms are packed to the **GLBSIG** and **NLGLBSIG** matrices.

The element stress and strain responses which are required by any user function constraints are also computed in this module. Those response values are stored into a relation entity to be used by user function evaluation utilities.

Design Requirements:

1. The **SMAT** (or **NLSMAT**), **GRIDTEMP** and **TREF** (or **TREFD**) entities must exist.

2. The **CASE** relation must be complete from SOLUTION.

Error Conditions:

1. A zero material allowable may cause division by zero in the computation of some of the constraints.

Engineering Application Module: **SCHDULER**

Entry Point: **AROSCH**

Purpose:

To determine the value of scheduled aerodynamic parameters, provide the values to the trim modules, and check for convergence of the scheduled parameters.

MAPOL Calling Sequence:

```
CALL SCHDULER ( BCID, SUB, TRIMDATA, TRIMRSLT, TRIMTOC,
                [AAR], [DELTA(SUB)], SCHITER, SCHCNVG );
```

BCID	User defined boundary condition identification number (Integer, Input)
SUB	Subscript number of **TRIM** subcases considered in this call (Integer, Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
TRIMRSLT	Relation containing **TRIMDATA** and results of the trim solutions (The schedule results are stored in this relation.) (Output)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aerodynamic loads matrices (Input)
[AAR]	Matrix of acceleration vectors (Input)
[DELTA(SUB)]	Matrix of configuration parameters (Input)
SCHITER	Schedule iteration number (First iteration = 1) (Integer, Input)
SCHCNVG	Schedule convergence indicator (Logical, Output)

Application Calling Sequence:

None

Method:

The module begins by bringing into memory the **CASE** entries associated with **TRIM** subcases in the current boundary condition. Then, the **TRIMTOC** relation is read into memory. The **TRIMDATA** and **TRIMRSLT** relations are read for all entries that have the current boundary identification and subscript value. The data from the **CONLINK** relation is also read into memory.

An evaluation of the trim data is done to determine the number of trim cases that will be processed during this pass (for the current subscript). Each trim subcase must be processed separately because of the options for control linking and scheduling. The first step is to determine which **TRIM** entries are associated with the current subcase (note all are associated with the current subscript). Once the **TRIM** identification of the current case is known, the **CASE** relation data are searched to determine the subcase number (1 to n over all **TRIM** entries in **CASE** for each **BCID**).

Each entry in the current **TRIM** case is checked for a scheduled parameter as indicated by the character string 'SCHD' in the 'FREEI' field. If a scheduled parameter is found, the value from the previous iteration is retrieved from the **TRIMRSLT** relation. The schedule data for the parameter is

read from the **SCHEDULE** and **AEFACT** relations. Now the values of the parameters that the schedule is a function of must be determined.

If the schedule references Mach number, dynamic pressure, or free stream velocity, these values are retrieved from the current **TRIM** tuple. If the schedule references a parameter listed on the **TRIM** entry, its value is retrieved from the **TRIMRSLT** relation. If the schedule references a parameter not on the **TRIM** entry, the **TRIMTOC** relation is searched to determined if the parameter value can be retrieved from the **AAR** or **DELTA(SUB)** matrices. Finally, the **CONLINK** relation is searched for referenced parameters not found in the previous searches. If a linked parameter is referenced by the schedule, its value is built up from the values of the source parameters in the **AAR** and **DELTA(SUB)** matrices.

After all the values of the referenced parameters are determined, the **MLTINT** routine is called to calculate the scheduled parameter's value at the current trim state. The schedule is checked for convergence by comparing the difference between the current and previous schedule values against a user defined tolerance. The new value of the scheduled parameter is stored on the **TRIMRSLT** relation and printed out if a **TRIM** print was requested. The scheduled parameter will be treated as a fixed value in the subsequent **TRIM** solution.

After all the trim cases in the current subscript have been processed, the convergence of all the schedules are checked to determine if another pass through the trim iteration loop is required. If no schedules were found, the convergence flag, **SCHCNVG**, is set to TRUE. If schedules were found, then every schedule within this subscript must converge for **SCHCNVG** to be set equal to TRUE. Convergence cannot occur on the first iteration in which schedules were calculated because the trim routines have not been called yet. After the convergence flag has been set, the module terminates.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **SFORLD**

Entry Point: **SFORLD**

Purpose:

To generate rigid structural trim parameter load vector group.

MAPOL Calling Sequence:

```
CALL SFORLD ( , GSIZEB, GLBDES, SMPLOD, [DPTHVI], [DPTHVD], [DPGRVI],
                [DPGRVD], RIGDSLOD, [SLPFRC]);
```

GSIZEB	Column dimension of [SLPFRC] matrix (Integer, Input)
GLBDES	Global design variable relation (Input)
SMPLOD	Unstructured entity of simple loads from LODGEN (Input)
[DPTHVI]	Matrix containing thermal load sensitivities from TELM structural domain in the F-set (Input)
[DPTHVD]	Matrix containing thermal load sensitivities from TELMD (Input)
[DPGRVI]	Matrix containing gravity load sensitivities from DMVI0 (Input)
[DPGRVD]	Matrix containing gravity load sensitivities from DMVID (Input)
RIGDSLOD	Group name and address of the current subscripted aerodynamic model user defined rigid structural loads (Input)
[SLPFRC]	Matrix containing the columns for user defined loads (Output)

Application Calling Sequence:

None

Method:

The **SLPARM** relation data are brought into memory to obtain the mechanical, thermal, and/or gravity simple load identification numbers for aeroelastic analyses that will use static load parameters for user defined control parameters. The **LOAD** bulk data relation is also read into memory to process combined simple loads requests as control parameters. The **SMPLOD** data are read to determine the number and types of each simple load defined in the bulk data packet. The **RIGDSLOD** relation is opened and prepared for generation of the **RIGSLOD** group. A loop is initiated over the number of simple loads. A unique column vector **SLPFRC** is prepared for each loop. In the loop, a search is conducted to determine if the load is :

1) a simple mechanical load,
2) a simple gravity load,
3) a simple thermal load, or
4) a combination of mechanical and/or gravity loads.

The column vector [SLPFRC] is assembled using the **SMPLOD** (and **LOAD**) data. The thermal and gravity loads are special in that the **GLBDES** information must be retrieved in order to assemble the loads representing the current design. The **SFORLD** module was developed from the **GTLOAD** module. The design portion of the simple loads and static loads parameters was not enabled in at this

time. The case where no design variables are defined does not represent a special case, however, since the [DPVRGI] and [DPTHGI] entities always include terms representing the "zeroth" design variable. Once all of the **SLPARM** entries have been processed, the module terminates.

Design Requirements:

The **SMPLD** entity from the **LODGEN** module must exist.

Error Conditions:

No simple loads are defined in the **SMPLOD** entity.

Engineering Application Module: **SOLUTION**

Entry Point: **SOLUTN**

Purpose:
To interpret the solution control packet.

MAPOL Calling Sequence:

```
CALL SOLUTION ( NUMOPTBC, NBNDCOND, K6ROT, MPS, MPE, FSDS, FSDE,
                MAXITER, MOVLIM, WINDOW, ALPHA, CNVRGLIM, NRFAC,
                EPS, FDSTEP, TOLVALUE );
```

NUMOPTBC	Number of optimization boundary conditions (Integer, Output)
NBNDCOND	Total number of optimization and analysis boundary conditions (Integer, Output)
K6ROT	Stiffness value for plate element "drilling" degrees of freedom (Real, Output)
MPS	The first iteration to use math programming (Integer, Output)
MPE	The last iteration to use math programming (Integer, Output)
FSDS	The first iteration to use FSD (Integer, Output)
FSDE	The last iteration to use FSD (Integer, Output)
MAXITER	The maximum number of allowable iterations (Integer, Output)
MOVLIM	Limit on how much a design variable can move for this iteration in using math programming (Real, Output)
WINDOW	The window around the zero in which the **MOVLIM** bound is overridden to allow the local variable to change sign. If **WINDOW** is zero the local variable may not change sign. If **WINDOW** is nonzero, the half width of a band around zero, **EPS** is computed **EPS = WINDOW/100 * MAX (ABS(TMIN), ABS(TMIN))** If the local variable falls within the band, the new minimum or maximum for the current iteration is changed to lie on the other side of zero from the local variable. The bandwidth EPS is a percentage of the larger of TMAX or TMIN where WINDOW specifies the percentage. (Real, Output)
ALPHA	Exponential move limit for the FSD algorithm (Real, Output)
CNVRGLIM	Relative percent change in the objective function that indicates approximate problem convergence (Real, Output)
NRFAC	Determines the minimum number of retained constraints equal to **NRFAC*NDV** (Real, Output)
EPS	A second criteria for constraint retention. All constraints greater than or equal to **EPS** will be retained (Real, Output)

`FDSTEP`	Relative design variable increment for finite difference computations (Real, Output)
`TOLVALUE`	Tolerance value for determine whether a node is on symmetry plane. (Real, Output)

Application Calling Sequence:

None

Method:

The SOLUTION module interprets the solution control statements and loads the resultant information to the **CASE** relation. On completion of the routine, the total number of all boundary conditions, the number of analysis boundary conditions and the user's optimization strategy are output to the executive sequence to direct the MAPOL execution path.

Design Requirements:

1. A Solution Control packet must be included in the input data stream.

Error Conditions:

1. Syntax errors and inconsistent or illegal solution control requests are flagged and the execution is terminated.

Engineering Application Module: **SPLINES**

Entry Point: **SPLINE**

Purpose:

Generates the interpolation matrices that relate displacements and forces between the structural and steady aerodynamic MODELs.

MAPOL Calling Sequence:

```
CALL SPLINES ( MODEL, GSIZEB, SAGEOM, SACOMPS, AEROS, [GTKG],
               [GSTKG], [GPTKG], [KREALK], [KIMAGK], [KIMAGS],
               [KIMAGP] );
```

MODEL	Name of steady aerodynamic input packet (Text, Input)
GSIZEB	The number of degrees of freedom in the set of all structural GRID and SCALAR points (Integer, Input)
SAGEOM	A relation describing the aerodynamic boxes for the steady aerodynamics model. The location of the box centroid, normal and pitch moment axis are given. It is used in splining the aerodynamics to the structure and to map responses back to the aerodynamics boxes. (Input)
SACOMPS	A relation describing aerodynamic components for the steady aerodynamics model. It is used in splining the aerodynamics to the structural model. (Input)
AEROS	A relation containing the definition of the aerodynamic coordinate system (Input)
[GTKG]	The matrix of splining coefficients relating the aeroelastic pressure increments to forces at the structural grids — the aeroelastic load increment spline (Output)
[GSTKG]	The matrix of splining coefficients relating the structural displacements to the streamwise slopes of the aerodynamic boxes — the aeroelastic slope increment spline (Output)
[GPTKG]	The matrix of splining coefficients relating the rigid aerodynamic pressures to forces at the structural grids — the aerodynamic load spline (Output)
[KREALK]	The matrix to transform the [GTKG], [GSTKG], [GPTKG] matrices to the real side for substructure reflection with QUADPAN method. (Text, Output)
[KIMAGK]	The matrix to transform the [GTKG] matrix to the image side for substructure reflection with QUADPAN method. (Text, Output)
[KIMAGS]	The matrix to transform the [GSTKG] matrix to the image side for substructure reflection with QUADPAN method. (Text, Output)

[KIMAGP]	The matrix to transform the [GPTKG] matrix to the image side for substructure reflection with QUADPAN method. (Text, Output)

Application Calling Sequence:

None

Method:

All the **SPLINE1**, **SPLINE2**, **SPLINE3** and **ATTACH** data are read and those associated with the steady aerodynamic model as described by the **SACOMPS** entity are used to assemble a list of aerodynamic boxes and structural grids for each spline. The **SAGEOM** relation is used to obtain the basic coordinates of the aerodynamic boxes and the **BGPDT** relation is used to obtain the locations of the structural grids. The spline matrix consisting of two columns (displacement and slope) for each aerodynamic box and 6 rows for each structural grid is then assembled for the aerodynamic boxes and structural grids attached to the spline.

The spline matrix is then expanded to include two columns for each aerodynamic box in the steady aerodynamic model and **GSIZEB** rows. It is then split into two pieces with each odd-numbered column (displacement) merged with previously processed splines to form the **GPTKG** and **GTKG** matrices and each even numbered (slope) column merged to form **GSTKG**. The process is repeated until all splines have been completed. The final matrices are returned to the MAPOL sequence.

Design Requirements:

None

Error Conditions:

1. Each aerodynamic box may appear on only one **SPLINE1**, **SPLINE2**, **SPLINE3** or **ATTACH** entry although not all boxes need appear. Missing boxes will not influence the aeroelastic response.

2. Missing structural grids or aerodynamic elements appearing on the spline definitions will be flagged.

Engineering Application Module: **SPLINFND**

Entry Point: **SPLFND**

Purpose:

To obtain the names of spline matrices for the specified boundary condition .

MAPOL Calling Sequence:

```
CALL SPLINFND (BCID, CASE, MODEL, SPLINE, [GTKG], [GSTKG], [GPTKG]);
```

BCID	Boundary condition identification number (Integer, Input)
CASE	Relation containing solution control case definition. (Text, Input)
MODEL	Name of steady aerodynamic model (Text, Input)
SPLINE	Group relation contains spline matrices (Text, Output)
[GTKG]	The matrix of splining coefficients relating the aeroelastic pressure increments to forces at the structural grids — the aeroelastic load increment spline (Output)
[GSTKG]	The matrix of splining coefficients relating the structural displacements to the streamwise slopes of the aerodynamic boxes — the aeroelastic slope increment spline (Output)
[GPTKG]	The matrix of splining coefficients relating the rigid aerodynamic pressures to forces at the structural grids — the aerodynamic load spline (Output)

Application Calling Sequence:

None

Method:

This module first obtains the aerodynamic model name for the specified boundary condition from the **CASE** relation. Then, **SPLINE** group name is obtained from the model group. The names for **SPLINE** group member matrices **GTKG, GSTKG, GPTKG** are also obtained.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **SPLNGNDR**

Entry Point: **SPLGDR**

Purpose:

To examine solution control steady aeroelastic cases and create SPLINE group and member entities.

MAPOL Calling Sequence:

```
CALL SPLNGNDR ( MINDEX, CASE, LOOP, MODEL, SAEMODEL, SAMODEL, SAGEOM,
                SACOMPS, SPLINE, [GTKG], [GSTKG], [GPTKG], FLEXLOAD,
                NEWMODEL, GOSPLINE, [KREALK], [KIMAGK], [KIMAGS],
                [KIMAGP] );
```

MINDEX	Mach number index for the current pass. (Integer, Input)
CASE	Relation containing solution control case definition. (Text, Input)
LOOP	Logical flag to indicate next pass is needed. (Logical, Output)
MODEL	Name of the current model. (Text, Output)
SAEMODEL	Group relation contains steady aeroelastic model group. (Text, Output)
SAMODEL	Group relation contains steady aerodynamic model group. (Text, Output)
SAGEOM	Group member entity contains aerodynamic geometry relation. (Text, Output)
SACOMPS	Group member entity contains aerodynamic components relation. (Text, Output)
SPLINE	Group relation contains spline matrices. (Text, Output)
[GTKG]	The matrix of splining coefficients relating the aeroelastic pressure increments to forces at the structural grids — the aeroelastic load increment spline (Output)
[GSTKG]	The matrix of splining coefficients relating the structural displacements to the streamwise slopes of the aerodynamic boxes — the aeroelastic slope increment spline (Output)
[GPTKG]	The matrix of splining coefficients relating the rigid aerodynamic pressures to forces at the structural grids — the aerodynamic load spline (Output)
FLEXLOAD	Group relation contains flexible trim parameter load vectors. (Text, Output)
NEWMODEL	Logical flag indicating new model. (Logical, Output)

`GOSPLINE`	Logical flag to generate spline matrices. (Logical, Output)
`[KREALK]`	The matrix to transform the `[GTKG]`, `[GSTKG]`, `[GPTKG]` matrices to the real side for substructure reflection with QUADPAN method. (Text, Output)
`[KIMAGK]`	The matrix to transform the `[GTKG]` matrix to the image side for substructure reflection with QUADPAN method. (Text, Output)
`[KIMAGS]`	The matrix to transform the `[GSTKG]` matrix to the image side for substructure reflection with QUADPAN method. (Text, Output)
`[KIMAGP]`	The matrix to transform the `[GPTKG]` matrix to the image side for substructure reflection with QUADPAN method. (Text, Output)

Application Calling Sequence:

None

Method:

The module begins by bringing the **CASE** entries associated with **FTRIM** subcases in the current boundary condition into memory. Then, for the specified input Mach index **MINDEX**, the boundary condition ID, subcase ID, aerodynamic model name, Mach number, the name of the method which creates the model, trim symmetry and **AIC** matrix symmetric option are obtained from **CASE** entries. The logical flag **NEWMDL** is then checked to determine if there is a new model for the current subcase. If it is a new model, and if the target model group is assembled from a solution control aerodynamic model assembly command, then the groups are fully or partially created by **IMPORT**, **ARCHIVE** or **OVERLAY** operations. Then, any related member groups and member entities are created if they were not previously created by solution control.

Group **SAEMODEL** is then created and copied from group **SAMODEL**, Group **SPLINE** and its member matrices are created, and finally Group **FLEXLOAD** is created.

Design Requirements:

None.

Error Conditions:

None

Engineering Application Module: `TRIMCHEK`

Entry Point: `TRMSOL`

Purpose:

To obtain the names of spline matrices for the specified boundary condition .

MAPOL Calling Sequence:

```
CALL TRIMCHEK ( CASE, TRIMDATA );
```

`CASE`	Relation contains solution control case definition. (Text, Input)
`TRIMDATA` information	Relation contains **TRIM** entry description and other **TRIM** such as Mach index, subscript number, model name, aero method. (Text, Output)

Application Calling Sequence:

None

Method:

The **TRIMCHEK** preface module performs initial aerodynamic processing for planar STEADY aerodynamics. It is driven by the **TRIM** data present in the bulk data packet and the SAERO disciplines in the **CASE** relation. The **CASE** relation provides the symmetries while the **TRIM** relation provides the Mach numbers. Only if SAERO disciplines are in **CASE** is any processing done and both **TRIM** and **AEROS** entries must be found.

On each call, the **PASSDF** submodule is called to determine the set of all Mach numbers and, for each Mach number, whether symmetric, antisymmetric or both boundary conditions are to be applied. Having determined all unique Mach numbers, the **PASSDF** then determines the **MINDEX'th** Mach number in numerical order (lowest to highest).

Relation **TRIMDATA** is created in this module. It has all **TRIM** entries along with **MINDEX**, aerodynamic model name and method name.

Design Requirements:

None

Error Conditions:

1. Errors in the **TRIM** specifications are flagged.

Engineering Application Module: **UDEFGEN**

Entry Point: **UDFGEN**

Purpose:

To assemble user defined load parameters for static aeroelastic analysis.

MAPOL Calling Sequence:

```
CALL UDEFGEN ( NITER, BCID, SUB, SYMTRN(BC), RIGDSLOD, [UDGFORCE],
               [UDFALOAD], GSIZE, TLABEL, TRIMDATA, STDYGEOM,
               TRIMTOC, MACH, YESUDEF );
```

NITER	Optimization iteration number (Integer, Input)
BCID	Boundary condition number (Integer, Input)
SUB	Subscript value (Integer, Input)
SYMTRN(BC)	Logical denoting the presence of a "full" structural model instantiated from a centerline symmetric structural model via substructuring techniques (Input)
RIGDSLOD	Group name and address of the current subscripted aerodynamic model user defined rigid structural loads (Input)
[UDGFORCE]	Matrix containing the columns for user defined loads in the structural domain in the g-set (Output)
[UDFALOAD]	Matrix containing the null columns for user defined loads in the aerodynamic domain. User defined loads can only be defined in the structural domain. (Output)
TLABEL	Unstructured entity containing a master label list of trim control effectors and associated symmetries for current subscript. Used in downstream processes to assemble user defined loads and aerodynamic forces as well as acceleration forces in this routine (Input)
TRIMDATA	Relation containing the **TRIM** Bulk Data and related boundary condition, subcase and subscript information (Input)
STDYGEOM	Group name and address of the current subscripted aerodynamic model geometry (Input)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aeroelastic load matrices (Input)
MACH	Aerodynamic Mach number for current subscript (Real, Input)
YESUDEF	Logical indicating the presence of user defined loads in the current subscript (Output)

Application Calling Sequence:

None

Method:

The routine creates a null vector column for user defined loads in the aerodynamic domain as well as an actual load vector column in the structural domain g-set for every user defined parameter. The **STDYGEOM** group of the current subscript is queried to determine the length the load column in the aerodynamic domain. **GSIZE** is passed in from MAPOL to provide the length of the load vector columns in the structural domain. The **TLABEL** entity is used to drive this routine.

A loop is established over the list of labels. Associated entities from the **RIGDSLOD** group are extracted. Individual load vectors are attained and packed one column at a time in the **UDGFORCE** matrix. Accordingly, a null vector is packed in the **UDFALOAD** matrix. As a vector is packed, associated column, label, and vector symmetry are recorded in the **TRIMTOC** relation. The process is continued until the list of trim parameter labels in the **TLABEL** entity is exhausted.

Design Requirements:

The **ACCFGEN** module must have been called prior to the **UDEFGEN** module.

Error Conditions:

None

Engineering Application Module: **UDEFTRAN**

Entry Point: **UDFTRN**

Purpose:

To transform user defined load parameters of static aeroelastic analysis from centerline symmetric structural representation to full structural model in the case of applied substructuring techniques.

MAPOL Calling Sequence:

```
CALL UDEFTRAN ( NITER, BCID, SUB, [UDFFORCE], TRIMTOC,
                [HFREALT(BC)], [HFIMAGT(BC)], [UDFFORCX] );
```

NITER	Optimization iteration number (Integer, Input)
BCID	Boundary condition number (Integer, Input)
SUB	Subscript value (Integer, Input)
[UDFFORCE]	Matrix containing the columns for user defined loads in the structural domain in the f-set (Input)
TRIMTOC	Relation containing the Trim Table Of Contents which describes the aeroelastic load matrices (Input)
[HFREALT(BC)]	Matrix containing transform operator for real side of structure in case of substructuring (Input)
[HFIMAGT(BC)]	Matrix containing transform operator for image side of structure in case of substructuring (Input)
[UDFFORCX]	Matrix containing the columns for user defined loads in the structural domain in the f-set for the full structure (Output)

Application Calling Sequence:

None

Method:

The routine operates on the f-set user defined loads in matrix **UDFFORCE** one column at a time to transform it to a full structure representation. The **TRIMTOC** relation is used to unpack the **UDFFORCE** matrix. A loop is executed over the columns of the matrix. Each column is packed into a temporary single-column matrix **(PF0)**, and based on the symmetry of the user defined loads (maintained in **TRIMTOC)**, the vectors are operated on to form a full structure representation as follows:

[PF2] = [HFREALT(BC)] *[PF0] + RSYM * [HFIMAGT(BC)] *[PF0]

where for a symmetric parameter	**RSYM = 1**
and for an antisymmetric parameter	**RSYM = -1**

Each column vector, **PF2**, is repacked into the new matrix **UDFFORCX.**

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **USSAERO**

Entry Point: **STEADY**

Purpose:

To process the linear boundary condition steady aerodynamics (preface to static aeroelasticity discipline).

MAPOL Calling Sequence:

```
CALL USSAERO ( MINDEX, MODEL, CASEID, MACH, SYM, AICSYM, SACOMPS,
               SAGEOM, REFPARAM, [AIC], [AAIC], [AIRFRC], RIGDALOD,
               AEROGRID, CAEROBOX );
```

MINDEX Mach number index for the current pass. (Integer, Input)

MODEL Name of the current model. (Text, Input)

CASEID Subcase identification number. (Integer, Input)

MACH Mach number. (Real, Input)

SYM Trim symmetry flag. (Integer, Input)

= - 1	Antisymmtric
= 0	Asymmetric
= 1	Symmetric

AICSYM AIC matrix symmetry option. (Integer, Input)

= - 2	None
= - 1	Antisymmtric
= 0	Asymmetric
= 1	Symmetric
= 2	Both symmetric and antisymmetric

SACOMPS Group member entity contains aerodynamic components relation. (Text, Output)

SAGEOM Group member entity contains aerodynamic geometry relation. (Text, Output)

REFPARAM Group member relation entity contains nominal trim parameter settings. (Text, Output)

[AIC] Group member entity contains the steady aerodynamic influence coefficient matrix for SYMmetric Mach numbers. (Text, Output)

[AAIC] Group member entity contains the steady aerodynamic influence coefficient matrix for ANTISYMmetric Mach numbers. (Text, Output)

RIGDALOD Group relation contains rigid aerodynamic trim parameter load. (Text, Output)

AEROGRID Group member entity contains aerodynamic grid geometry relation. (Text, Output)

CAEROBOX	Group member entity contains aerodynamic connection relation. (Text, Output)

Application Calling Sequence:

None

Method:

The USSAERO preface module performs initial aerodynamic processing for planar STEADY aerodynamics. It is driven by the **TRIMDATA** relation and the **MINDEX** value.

On each call, the **TRIMDATA** relation is queried to determine the MINDEX'th Mach number and whether symmetric, antisymmetric or both boundary conditions are to be applied.

On the first call (determined by **MINDEX=1**) the STEADY module computes the planar STEADY aerodynamic geometry in calls to **GEOM**. It then processes the current Mach number and stores the resultant **AIC** terms in the **AICMAT** and/or **AAICMAT** entity (depending on the symmetry options) and in the resultant rigid forces in the **AIRFRC** matrix.

Design Requirements:

None

Error Conditions:

1. Errors in the STEADY aerodynamic **MODEL** specifications are flagged.

THIS PAGE INTENTIONALLY LEFT BLANK.

4. DATABASE ENTITY DESCRIPTIONS

This section presents updates to Chapter 9 of the ASTROS Programmer's Manual (Reference 5). Included are modifications to the database entity descriptions. These are summarized in the following table:

ENTITY	TYPE	DESCRIPTION
ACCELOAD	Matrix	Matrix containing the null columns for acceleration loads in the aerodynamic domain
ACCFORCE	Matrix	Matrix containing the null columns for acceleration loads in the structural domain
AEROLOAD	Matrix	Matrix containing the columns for aerodynamic loads in the aerodynamic domain
AIC	Group	Contains steady aerodynamic AIC matrices
ARCHIVE	Relation	Contains definitions for aero model assembly command ARCHIVE
ASSEMBLE	Relation	Contains definitions for aero model assembly command ASSEMBLE
BMST	Relation	Contains the component load (BMST) definitions in the structural domain
BMST2	Relation	Contains the component load (BMST) definitions in the aerodynamic domain
BMSTDATA	Relation	Contains the trim parameter component load (BMST) data
CASE	Relation	Contains processed Solution Control commands
CONST	Relation	Contains the constraint values and constraint sensitivity processing data
CONSTORD	Relation	Contains the reordered constraint values and constraint sensitivity data for the current design iteration
DCONBMST	Relation	Contains the component load (BMST) constraints to be used in a design optimization
DCONSCF	Relation	Contains the definition of a constraint on the flexible stability derivative at the reference grid point associated with the force or moment due to a trim parameter or control surface deflection of the form
DRAGDATA	Relation	Contains the trim parameter drag data
FLEXLOAD	Group	Contains flexible trim parameter load data
IMPORT	Relation	Contains definitions for aero model assembly command IMPORT
OBMSTLOD	Relation	Contains the trimmed BMST component loads data including rigid, rigid-spline, flexible increment, and applied data
OVERLAY	Relation	Contains definitions for aero model assembly command OVERLAY
PANLST1	Relation	Contains aerodynamic panel set definitions by identifying opposite corner panels
PANLST2	Relation	Contains aerodynamic panel set definitions by a list
REFPARAM	Relation	Contains nominal trim parameter settings
RELES	Relation	Contains list of grid point degrees of freedom to be released during substructure operation
RIGDALOD	Group	Contains rigid aerodynamic trim parameter load data
RIGDSLOD	Group	Contains user defined rigid structural load data
RMASS	Relation	Contains the rigid mass for the rigid trim discipline, RTRIM
SAEMODEL	Group	Contains steady aeroelastic model definition
SAMODEL	Group	Contains steady aerodynamic model definition

SAROLOAD	Matrix	Matrix containing the incremental non-dimensional aerodynamic load at the aerodynamic panel control points for the current Model / Method /Mach combination (Force/QDP)
SCHEDULE	Relation	Contains the trim parameter schedule definitions
SLPARM	Relation	Static Load Trim Parameter.
SLPFRC	Group	Matrix containing the columns for user defined loads
SPLINE	Group	Contains steady aerodynamic model to structural model spline matrices
STABCFA	Relation	Stability Coefficients In Aerodynamic Domain.
STABCFS	Relation	Stability Coefficients In Structural Domain.
STDYGEOM	Group	Contains steady aerodynamic model geometry group definition
TCONBMST	Relation	Contains the component load (BMST) constraints for the trim optimization problem
TCONFUNC	Relation	Contains the functional constraints for the trim optimization problem
TCONTRM	Relation	Contains the trim parameter constraints for the trim optimization problem
TFUNC	Relation	Contains weighted sum function definitions to be used in trim optimization
TLABEL	Unstructured	Master label list for development of TRIMTOC and Loads for steady aeroelastic analysis
TODVPRM	Relation	Contains the trim optimization design variables
TOMPPARM	Relation	Contains the optimizer parameters and their new values for use in the trim mathematical programming optimizer
TRIM	Relation	Contains the specified conditions for steady aeroelastic trim as input from Bulk Data file.
TRIMDATA	Relation	Contains the TRIM Bulk Data and related boundary condition, subcase, and subscript information
TRIMOPT	Relation	Contains the trim optimization problem specification
TRIMR	Relation	Contains the specified conditions for steady rigid aerodynamic trim as input from Bulk Data file.
TRIMRSLT	Relation	Contains the TRIM Bulk Data, related boundary condition, subcase, and subscript information, and the results of the trim solutions.
TRIMTOC	Relation	Trim Parameter Table of Contents - Pointer to PAF.
UDFALOAD	Matrix	Matrix containing the null columns for user defined loads in the aerodynamic domain.
UDFFORCE	Matrix	Matrix containing the columns for user defined loads in the structural domain in the n-set
UDFFORCX	Matrix	Matrix containing the columns for user defined loads in the structural domain in the f-set for the full structure
UDGFORCE	Matrix	Matrix containing the columns for user defined loads in the structural domain in the g-set
UDNFORCE	Matrix	Matrix containing the columns for user defined loads in the structural domain in the n-set

Entity: **ACCELOAD**

Entity Type: Matrix

Description: Matrix containing the null columns for acceleration loads in the aerodynamic domain

Matrix Form: Rectangular and real. The number of rows is equal to the number of panels in the steady aerodynamics model and the number of columns is equal to the number of acceleration (ACCE) parameters in TRIMTOC.

Created By: Module ACCFGEN

Notes:

1. Refer to ACCFGEN module documentation for more information.

Entity: **ACCFORCE**

Entity Type: Matrix

Description: Matrix containing the null columns for acceleration loads in the structural domain

Matrix Form: Rectangular and real. The number of rows is equal to the number of degrees-of-freedom in the f-set and the number of columns is equal to the number of acceleration (ACCE) parameters in TRIMTOC.

Created By: Module ACCFGEN

Notes:

1. Refer to ACCFGEN module documentation for more information.

Entity: **AEROLOAD**

Entity Type: Matrix

Description: Matrix containing the columns for aerodynamic loads in the aerodynamic domain

Matrix Form: Rectangular and real. The number of rows is equal to the number of panels in the steady aerodynamics model and the number of columns is equal to the number of aerodynamic (AERO + BASE) parameters in TRIMTOC.

Created By: Module ARFMRG

Notes:

1. Refer to ARFMRG module documentation for more information.

Entity: **AIC**

Entity Type: Relation

Description: Steady aerodynamic AIC matrices

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
MODELTYP	C(4)	Type of steady aerodynamic model
AEROMETH	C(8)	Name of method which created AIC matrices
MACH	RSP	Mach number
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module AROGNDRV

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

Entity: **ARCHIVE**

Entity Type: Relation

Description: Aero model assembly command ARCHIVE definition

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
NEWGNAME	C(24)	New name of the GROUP to be archived
GNAME	C(24)	Old name of the GROUP to be archived
GTYPE	C(12)	Type of the GROUP to be archived (STDYGEOM, AIC ...)
DBNAME	C(24)	Logical database name

Created By: Module SOLUTION

Entity: **ASSEMBLE**

Entity Type: Relation

Description: Aero model assembly command ASSEMBLE definition

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MODELNAM	C(8)	Name of the resulting model
MTYPE	C(12)	Type of model to be assembled (SAMODEL, SAEMODEL)
GLIST	INT	Integer identification number referencing the Unstructured entity GRPLIST which defines the name and type of groups to be assembled

Created By: Module SOLUTION

Entity: **BMST**

Entity Type: Relation

Description: Contains the component load (BMST) definitions in the structural domain

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
BMSTLAB	Text(8)	Label of the component load
CID	Integer > 0	Reference coordinate system identification number
GID	Integer > 0	Reference grid point identification number
GLSTID	Integer > 0	Grid list set identification number
PLSTID	Integer > 0	Aerodynamic panel list set identification number

Created By: Module IFP

Entity: **BMST2**

Entity Type: Relation

Description: Contains the component load (BMST) definitions in the aerodynamic domain

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
BMSTLAB	Text(8)	Label of the component load
PLSTID	Integer > 0	Aerodynamic panel list set identification number
X1, Y1, Z1	Real	Coordinates of the origin of the reference coordinate system
X2, Y2, Z2	Real	Coordinates of the z-axis of the reference coordinate system
X3, Y3, Z3	Real	Coordinates in the x-z plane of the reference coordinate system
XREF, YREF, ZREF	Real	Coordinates of the reference point at which the component load is defined

Created By: Module IFP

Notes:

1. Refer to the **BMST2** Bulk Data Entry for further information.

Entity: **BMSTDATA**

Entity Type: Relation

Description: Contains the trim parameter component load (BMST) data

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
BMSTLAB	Text(8)	Component load (BMST) label
LOADTYPE	Text(8)	Load type (RIGID, RSPLINED, or FLEXIBLE)
IMAGE	Integer	Flag denoting loads on image side, if substructuring (0 : False - Real Side, 1 : True - Image Side)
MODELNAM	Text(8)	Name of steady aerodynamic input packet
METHOD	Text(8)	Name of method which created aerodynamic loads
MACH	Real > 0.0	Mach number
SYMFLAG	Integer	Trim parameter symmetry (-1 : ANTI, 0 : ASYM, 1 : SYM)
PARAM1	Text(8)	Name of first level trim parameter
PRMVAL1	Real	Value of first level trim parameter
PARAM2	Text(8)	Name of second level trim parameter
PRMVAL2	Real	Value of second level trim parameter
PARAM3	Text(8)	Name of third level trim parameter
PRMVAL3	Real	Value of third level trim parameter
FX, FY, FZ	Real	Longitudinal, lateral, and vertical force on component
MX, MY, MZ	Real	Rolling, pitching, and yawing moment on component

Created By: Module PARMBMST

Notes:

1. The flexible trim parameter component load (BMST) data is used by the trim optimization modules when a BMST is specified as an objective or constraint. The rigid and rigid-splined BMST data is used to evaluate the spline on a component basis.

Entity: **CASE**

Entity Type: Relation

Description: Contains the CASE parameters for each analysis within each boundary condition as input in the solution control packet.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
OAFLAG	I>0	Optimize/analyze flag 1 Optimize 2 Analyze
BCID	I>0	Boundary condition identification number
CASEID	KI>0	Subcase identification number
MPCSETID	I	Multipoint constraint set identification number
SPCSETID	I	Single point constraint identification number
AUTOSPC	I	AUTOSPC request flag
ASPCPRNT	I	AUTOSPC print flag
ASPCPUNC	I	AUTOSPC punch flag
ASPCEPS	I	AUTOSPC threshold
REDSETID	I	Guyan reduction constraint set identification number
SUPSETID	I	Support set identification number
METHOD	I	Real eigenvalue extraction set identification number
CMETHOD	I	Complex eigenvalue extraction set identification number
DYNRED	I	Dynamic reduction set identification number
INERTIA	I	Inertia relief mode shapes set identification number
TFSETID	I	Transfer function set identification number
K2PP	C(8)	K2PP matrix entity name
M2PP	C(8)	M2PP matrix entity name
B2PP	C(8)	B2PP matrix entity name
K2GG	C(8)	K2GG matrix entity name
M2GG	C(8)	M2GG matrix entity name

DISFLAG	I	Discipline flag 1 Statics 2 Modes 3 Steady Aero (Trim) 4 Flutter 5 Transient Response 6 Frequency Response 7 Buckling 8 Blast 9 Nonplanar Steady Aero 10 Static Aero (Generation)
MECHLOAD	I	Mechanical load set identification number
THRMLOAD	I	Thermal load set identification number
GRAVLOAD	I	gravity load set identification number
TRIMID	I	Trim set identification number
SAROSYM	I	Steady aerodynamic symmetry flag
TRIMOPT	I	Trim optimization identification number
DCONST	I	Design constraint set identification number
DCSTRESS	I	Stress constraint set identification number
DCSTRAIN	I	Strain constraint set identification number
DCONFID	I	DCONF identification number
DLOADID	I	Dynamic load set identification number
DRMETH	I	Dynamic response method 1 Direct 2 Modal
TIMESTEP	I	Time step set identification number
FREQSTEP	I	Frequency step set identification number
FFTID	I	Fast Fourier transform set identification number
GUSTID	I	Gust set identification number
INITCON	I	Initial condition set identification number
RANDOMID	I	Random set identification number
BLASTID	I	Blast set identification number
BUCKLEID	I	Buckling eigenvalue extraction set identification number
FLUTID	I	Flutter set identification number
CONTROL	C(8)	Name of aerodynamic extra point splining matrix
DAMPID	I	Damping set identification number
ESET	I	Extra point set identification number
SAROMETH	C(8)	The name of method which creates aerodynamic model
SAROMACH	RSP	Mach number
SAROMODL	C(8)	Steady aerodynamic model name
AICSYM	I	[AIC] matrix symmetry flag

RIGDCOND	I	Identification number for rigid stability coefficient computation condition
FLEXCOND	I	Identification number for flexible stability coefficient computation condition
SLPRMSET	I	Identification number to reference SLPARM bulk data
USESYMM	I	Identification number to reference the symmetry subcase by USING option in TRIM solution
USEANSYM	I	Identification number to reference the anti-symmetry subcase by USING option in TRIM solution
RELESID	I	Identification number to reference RELES bulk data
ACCEPRNT	I(20)	Acceleration print selector (1) Print set identification number>0 or: 0 None -1 ALL -2 LAST (2) Punch set identification number (3) Print form 0 Rectangular 1 Polar (4) Punch form (5) Print frequency set identification number (6) Punch frequency set identification number (7) Print iteration set identification number (8) Punch iteration set identification number (9) Print mode set identification number (10) Punch mode set identification number (11) Print time set identification number (12) Punch time set identification number
AIRDPRNT	I(20)	Aerodynamic displacement print selection
BUCKPRNT	I(20)	Buckling print selection
DISPPRNT	I(20)	Displacement print selection (1) Print set identification number (2) Punch set identification number (3) Print form 0 Rectangular 1 Polar (4) Punch form
ENERPRNT	I(20)	Strain energy print selection
FORCPRNT	I(20)	Element force print selection
GPFOPRNT	I(20)	Grid point force print selection
GPWGPRNT	I(20)	Grid point weight generation print selection
LOADPRNT	I(20)	Load print selection

MASSPRNT	I(20)	Mass matrix print selection
MPCFPRNT	I(20)	Multi-point constraint force print selection
QHHPRNT	I(20)	QHH matrix print selection
QHJPRNT	I(20)	QHJ matrix print selection
ROOTPRNT	I(20)	Flutter and normal modes eigenvalue print selection
SPCFPRNT	I(20)	Single point constraint force print selection
STIFPRNT	I(20)	Stiffness matrix print selection
STRAPRNT	I(20)	Strain print selection (13) Print layer/laminate option (14) Punch layer/laminate option 0 Layer strain 1 Laminate strain 2 Both layer and laminate strain
STREPRNT	I(20)	Stress print selection (13) Print layer/laminate option (14) Punch layer/laminate option 0 Layer strain 1 Laminate strain 2 Both layer and laminate strain
TPREPRNT	I(20)	Trim pressure coefficient print selection
VELOPRNT	I(20)	Velocity print selection
TRIMPRNT	I(20)	Steady aeroelastic trim print selection
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module SOLUTION

Notes:

1. The format of the ACCEPRNT vector is typical of the format of all the print selection vector. Additionally, the format for the print set. Additionally, the format for the print set identification numbers in the ACCEPRNT vector is typical of that of the other set identification numbers in the vector.

2. The CASE, JOB and OPTIMIZE relational entities together contain the solution control requests as input in the Solution Control packet. CASE contains the case-dependent parameters,JOB contains the case-independent requests, and OPTIMIZE contains the optimization dependent requests.

Entity: **CONST**

Entity Type: Relation

Description: Contains the constraint values and constraint sensitivity processing data.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
NITER	I>0	Iteration number
REFLECT	I = 0 or 1	Integer flag indicating substructure reflection = 0 Real side = 1 Image side
CVAL	R	Constraint value
CTYPE	I>0	Constraint type (see Note 2 below)
BCID	I>0 or NULL	Boundary condition identification number for constraint value if boundary condition dependentNon-boundary dependent constraints are: minimum thickness (CTYPE=1) maximum thickness (CTYPE=2) laminate composition (CTYPE=13) laminate min.gauge (CTYPE=14) ply min.gauge (CTYPE=15) BAR dimension relation (CTYPE=18)
DISFLAG	I>0 or NULL	Discipline type flag from CASE relation (where appropriate) Non-discipline dependent constraints are: minimum thickness (CTYPE=1) maximum thickness (CTYPE=2) laminate composition (CTYPE=13) laminate min.gauge (CTYPE=14) ply min.gauge (CTYPE=15) BAR dimension relation (CTYPE=18)
ACTVFLAG	I>0 or NULL	Flag denoting status of the constraint as active (=1) or inactive (=0) ACTVFLAG will have NULL value prior to constraint screening in ACTCONThe entry with CLASS as OBJECTIVE has its ACTVFLAG set to be active
SCNUM	I>0 or NULL	See Remark 11
PNUM	I>0 or NULL	See Remark 12
SUBSCRPT	I>0 or NULL	Subscript number for SAERO discipline constraints of types 3, 4, 5, 6, 9, 10, 11, and 12

DISPCOL	I>0 or NULL	Column number in the matrix of pseudo-displacements or accelerations for static aeroelastic constraints of types 9, 10, and 12
ETYPE	C(8) or NULL	Element type used for stress/strain and thickness constraints
EID	I or NULL	Element identification number
LAYERNUM	I or NULL	Element layer information (See Remark 14)
DVSYMBL	C(8)	Designed dimension symbol selected fro D1 through D10 for BAR element cross section dimension; A for element area, T for element thickness.
SCON	I>0 or NULL	See Remark 13
VSCON	R(6)	Allowables for stress/strain constraints
SENSPRM1	R	General values useful for sensitivity calculations (see Remark 10)
SENSPRM2	R	
SENSPRM3	R	
SENSPRM4	C(8)	
SENSPRM5	R(10)	
PRINTKEY	I	Pointer to the GRADIENT entity containing the gradient of the constraint with respect to the global variables, 0 if no gradient was stored (see Remark 15)
BENDPOST	I	Bending position flag for nonlinear sensitivity
INSTANCE	I	User function instance number
CLASS	C(16)	User function class (OBJECTIVE/CONSTRAINTS)
UFNAME	C(8)	User function name

Created By: See Note below.

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

4. Other notes in the current manual remain unchanged.

Entity: **CONSTORD**

Entity Type: Relation

Description: Contains the reordered constraint values and constraint sensitivity data for the current design iteration.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
NITER	I>0	Iteration number
REFLECT	I = 0 or 1	Integer flag indicating substructure reflection = 0 Real side = 1 Image side
CVAL	R	Constraint value
CTYPE	I>0	Constraint type (see Note 2 for CONST)
BCID	I>0 or NULL	Boundary condition identification number for constraint value if boundary condition dependent. Non-boundary dependent constraints are: minimum thickness (CTYPE=1) maximum thickness (CTYPE=2)
DISFLAG	I>0 or NULL	Discipline type flag from CASE relation (where appropriate) Non-discipline dependent constraints are: minimum thickness (CTYPE=1) maximum thickness (CTYPE=2) laminate composition (CTYPE=13) laminate min.gauge (CTYPE=14) ply min.gauge (CTYPE=15)
SCNUM	I>0 or NULL	See Remark 11 for CONST
PNUM	I>0 or NULL	See Remark 12 for CONST
SUBSCRPT	I>0 or NULL	Subscript number for SAERO discipline constraints of types 3, 4, 5, 6, 9, 10, 11, and 12
SCON	I>0 or NULL	See Remark 13 for CONST
SENSPRM4	C(8)	String for sensitivity
PRINTKEY	I	Pointer to the GRADIENT entity containing the gradient of the constraint with respect to the global variables= 0 if no gradient was stored (see Remark 15 for CONST)

Created By: Module IFP

Notes:

1. Refer to CONVERT Bulk Data entry for the valid QUANTITY values.

Entity: **DCONBMST**

Entity Type: Relation

Description: Contains the component load (BMST) constraints to be used in a design optimization

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	Integer > 0	Set identification number
BMSTLAB	Text(8)	Label of a component load
CMPNT	Integer	Component number (DOF) of BMST (+/- 1, 2, 3, 4, 5, or 6)
CTYPE	Text(8)	Constraint type (UPPER of LOWER)
BMSTLIM	Real	Component load bound

Created By: Module IFP

Notes:

1. A negative **CMPNT** refers to the image side when substructuring.

Entity: **DCONSCF**

Entity Type: Relation

Description: Contains the definition of a constraint on the flexible stability derivative at the reference grid point associated with the force or moment due to a trim parameter or control surface deflection of the form.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	I>0	Constraint set identification
ACCLAB	C(8)	Structural acceleration label
PRMLAB	C(8)	Constrained control surface label or aeroelastic trim parameter
CTYPE	C(8)	Constraint type
PRMREQ	R	Stability coefficient bounds
UNITS	C(8)	Stability coefficient units
FSCFLAG	R	Flag signifying flexible stability coefficient definition

Created By: Module IFP

Entity: **DRAGDATA**

Entity Type: Relation

Description: Contains the trim parameter drag data

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MACHINDX	Integer	Mach number index for the current subcase
SUBSCRPT	Integer	Subscript counter
RIGFLX	Integer	Rigid or Flexible Flag
PARM	Text(8)	Label defining the aerodynamic trim parameter
SYMFLG	Integer	Symmetry Flag
PARMVAL	Real	Parameter value
FX, FY, FZ	Real	Longitudinal, Lateral, and Vertical Force
MX, MY, MZ	Real	Rolling, Pitching, and Yawing Moment

Created By: Module DRAG

Notes:

1. Not Implemented

Entity: **FLEXLOAD**

Entity Type: Relation

Description: Flexible trim parameter load vectors

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
METHOD	C(8)	Name of method which created AIC matrices
LOADTYPE	C(8)	Type of load vectors (APPLIED, INERTIA, or STRUCTUR)
MACH	RSP	Mach number
QDP	RSP	Dynamic pressure
V0	RSP	Free stream velocity
SYM	C(4)	Trim parameter symmetry (SYM, ANTI or ASYM)
PARAM1	C(8)	Name of first level trim parameter
VAL1	RSP	Value of first level trim parameter
PARAM2	C(8)	Name of second level trim parameter
VAL2	RSP	Value of second level trim parameter
PARAM3	C(8)	Name of third level trim parameter
VAL3	RSP	Value of third level trim parameter
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module SPLNGNDR

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

4. An unlimited number of FLXFRC and FLXDEF MEMBERs are allowed, but MEMBERs must be a unique combination of the SYM, PARAM1, VAL1, PARAM2, VAL2, PARAM3 and VAL3 attributes.

5. The FLEXLOAD group contains data for only a single Mach number. Multiple Mach numbers require multiple groups.

Entity: **IMPORT**

Entity Type: Relation

Description: Aero model assembly command IMPORT definition

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
NEWGNAME	C(24)	New name of the GROUP to create
GNAME	C(24)	Old name of the GROUP to be imported
GTYPE	C(12)	Type of the GROUP to be imported (STDYGEOM, AIC ...)
DBNAME	C(24)	Logical database name

Created By: Module SOLUTION

Entity: **OBMSTLOD**

Entity Type: Relation

Description: Contains the trimmed BMST component loads data including rigid, rigid-splined, flexible increment, and applied data

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
NITER	Integer	Iteration number
BCID	Integer	The boundary condition number
DISC	Integer	Discipline type flag from CASE
SUBCASE	Integer	Subcase identification number
BMSTLAB	Text(8)	BMST component load label
LOADTYP	Text(8)	Label identifying the type of load (See Remark 1)
MODELNAM	Text(8)	Name of steady aerodynamic input packet
METHOD	Text(8)	Name of method which created aerodynamic loads
MACH	Real	Mach number
FX, FY, FZ	Real	Longitudinal, lateral, and vertical force on component
MX, MY, MZ	Real	Rolling, pitching, and yawing moment on component
IMAGE	Integer	Flag denoting loads on image side, if substructuring (0 : False - Real Side, 1 : True - Image Side)

Created By: Module OFPBMST

Notes:

1. The LOADTYPE is a textual key that identifies the load terms. The following values are used:
   - RIGID - Trimmed rigid aerodynamic loads in the aerodynamic domain
   - RSPLINED - Trimmed rigid-splined aerodynamic loads in the structural domain
   - FLEXINCR - Trimmed flexible increment to aerodynamic loads in the structural domain (FLEXINCR = APPLIED - RSPLINED)
   - APPLIED - Total applied structural load at the trimmed condition

Entity: **OVERLAY**

Entity Type: Relation

Description: Aero model assembly command OVERLAY definition

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
NEWGNAME	C(24)	Name of the resulting model
GTYPE	C(12)	Type of model to be overlayed
GLIST	INT	Integer identification number referencing an Unstructured entity GRPLIST which defines the name and type of groups to be overlayed

Created By: Module SOLUTION

Entity: **PANLST1**

Entity Type: Relation

Description: Defines a set of aerodynamic panels by identifying opposite corner panels

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	INT	Set identification number
MACROID	INT	Identification number of aerodynamic macroelement
BOX1	INT	The identification number of the first box on the aerodynamic macroelement
BOX2	INT	The identification number of the last box on the aerodynamic macroelement

Created By: Module IFP

Entity: **PANLST2**

Entity Type: Relation

Description: Defines a set of aerodynamic panels by a list

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	INT	Set identification number
MACROID	INT	Identification number of aerodynamic macroelement
BOXI	INT	The identification number of the aerodynamic box associated with macroelement MACROID

Created By: Module IFP

Entity: **REFPARAM**

Entity Type: Relation

Description: Nominal trim parameter settings

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
PARAM	C(8)	Name of trim parameter
VALUE	RSP	Trim parameter setting

Created By: Module AROGNDRV

Notes:

1. An unlimited number of entries may be used to describe the nominal vehicle configuration, however each PARAM name must be unique.

2. Only trim parameters with nominal settings other than 0.0 need be entered in the REFPARAM relation. If a trim parameter name is not included in the relation, it is assumed to have a nominal value of zero.

Entity: **RELES**

Entity Type: Relation

Description: Defines sets of component degrees of freedom at substructure GRID points which are not to be connected during a substructure operation.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SID	INT	Set identification number
SNAME	Text(8)	Basic substructure name
GID	INT	GRID or SCALAR point identification number
COMP	INT	Degree of freedom to be released

Created By: Module IFP

Entity: **RIGDALOD**

Entity Type: Relation

Description: Rigid aerodynamic trim parameter load vectors

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
MODELTYP	C(4)	Type of steady aerodynamic model
AEROMETH	C(8)	Name of method which created load vectors
MACH	RSP	Mach number
V0	RSP	Dynamic pressure
SYM	C(4)	Trim parameter symmetry (SYM, ANTI or ASYM)
PARAM1	C(8)	Name of first level trim parameter
VAL1	RSP	Value of first level trim parameter
PARAM2	C(8)	Name of second level trim parameter
VAL2	RSP	Value of second level trim parameter
PARAM3	C(8)	Name of third level trim parameter
VAL3	RSP	Value of third level trim parameter
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module AROGNDRV

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. The REFPARAM member entry defines the name of the basis rigid load vector on the PARAM1 attribute. VAL1 is defined to be 1.0 as the nominal trim parameter settings are defined by an unlimited number of entries in the REFPARAM relation. PARAM2 and PARAM3 are blank by definition.

4. An unlimited number of AIRFRC MEMBERs are allowed, but MEMBERs must be a unique combination of the SYM, PARAM1, VAL1, PARAM2, VAL2, PARAM3 and VAL3 attributes.

5. The data stored in the AIRFRC matrices are the actual loads produced by the specified parameter settings including that of the REFPARAM rigid load vector(e.g., thickness, camber, twist, onset angles, etc.). This is a change from the baseline ASTROS paradigm in which the parameter VAL and corresponding load vector represent the rigid load increment from the THKCAM reference vector rather than the actual load vector for the specified trim parameter settings.

6. The RIGDALOD group contains data for only a single Mach number. Multiple Mach numbers require multiple groups.

Entity: **RIGDSLOD**

Entity Type: Relation

Description: User defined rigid structural load vectors

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL
SYM	C(4)	Trim parameter symmetry (SYM, ANTI or ASYM)
PARMTYP	C(8)	Static load parameter type specified by SPLARM bulk data entry
PARAM1	C(8)	Name of first level trim parameter
VAL1	RSP	Value of first level trim parameter
PARAM2	C(8)	Name of second level trim parameter
VAL2	RSP	Value of second level trim parameter
PARAM3	C(8)	Name of third level trim parameter
VAL3	RSP	Value of third level trim parameter

Created By: Module AROGNDRV

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

4. An unlimited number of UDEFPRM MEMBERs are allowed, but MEMBERs must be a unique combination of the SYM, PARAM1, VAL1, PARAM2, VAL2, PARAM3 and VAL3 attributes.

Entity: **RMASS**

Entity Type: Relation

Description: Contains the rigid mass for the rigid trim discipline, RTRIM

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
RMID	Integer > 0	Rigid mass identification number
MASS	Real	Mass value
I11, I21, I22	Real	Mass moments of inertia
I31, I32, I33	Real	Mass moments of inertia

Created By: Module IFP

Entity: **SAEMODEL**

Entity Type: Relation

Description: Steady aeroelastic model group

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
METHOD	C(8)	Name of method which created aeroelastic model
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

Entity: **SAMODEL**

Entity Type: Relation

Description: Steady aerodynamic model group

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
METHOD	C(8)	Name of method which created aerodynamic model
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module AROGNDRV

Notes:

1. If solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

Entity: **SAROLOAD**

Entity Type: Matrix

Description: Matrix containing the incremental non-dimensional aerodynamic load at the aerodynamic panel control points for the current Model / Method / Mach combination (Force/QDP)

Matrix Form: Rectangular and real. The number of rows is equal to the number of panels in the steady aerodynamics model and the number of columns is equal to the number of parameters in TRIMTOC.

Created By: MAPOL

Notes:

1. SAROLOAD is created by appending the ACCELOAD, UDFALOAD, and AEROLOAD matrices, in that order.

Entity: **SCHEDULE**

Entity Type: Relation

Description: Contains the trim parameter schedule definitions

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
TRIMID	Integer > 0	Trim set identification number
SURFLAB	Text(8)	Label of trim parameter to be scheduled
CTOL	Real > 0.0	Convergence tolerance of scheduled trim parameter
MAXITER	Integer > 0	Maximum number of schedule iterations allowed
SVID	Integer > 0	Identification number of an AEFACT entry which contains the schedule values
PRMLABI	Text(8)	Label of a trim parameter of which the schedule is a function
PVIDI	Integer > 0	Identification number of an AEFACT entry which contains the parameter values at which the schedule values are defined

Created By: Module IFP

Notes:

1. Refer to the **SCHEDULE** Bulk Data Entry for further information.

Entity: **SLPARM**

Entity Type: Relation

Description: Static Load Trim Parameter

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
PARAM1	Text(8)	User defined control effector label
VAL1	RSP	Scale parameter for load combination
PARAM2	Text(8)	User defined control effector label (See Note 1)
VAL2	RSP	Scale parameter for load combination
PARAM3	Text(8)	User defined control effector label (See Note 1)
VAL3	RSP	Scale parameter for load combination
SYM	Text(4)	Trim parameter symmetry (SYM, ANTI, or ASYM)
SCALE	RSP	Scale parameter of component load within load combination
LOAD	Integer	Load set ID of FORCE, MOMENT, FORCE1, MOMENT1, PLOAD, GRAV load type (See Note 2).
PARMTYP	Text(4)	Load type (MECH, GRAV, THRM)

Created By: IFP

Notes:

1. PARAM2 and PARAM3 are used only for documentation at this release. Later implementation will provide for nonlinear trim.

2. Load set ID must be greater than zero.

Entity: **SLPFRC**

Entity Type: Matrix Group Member

Description: Matrix containing the columns for user defined loads

Matrix Form: Rectangular and real. The number of rows is equal to the number of degrees-of-freedom in the g-set and the number of columns is equal to the number of user-defined load parameters.

Created By: Module SFORLD

Notes:

1. Refer to the SFORLD module documentation for more information.

Entity: **SPLINE**

Entity Type: Relation

Description: Steady aerodynamic model to structural model spline matrices

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
AEROMETH	C(8)	Name of method which created aerodynamic model
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module SPLNGNDR

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

Entity: **STABCFA**

Entity Type: Relation

Description: Stability Coefficients In Aerodynamic Domain.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MODELNAM	Text(8)	Name of steady aerodynamic input packet
METHOD	Text(8)	Name of method which created load vector
MACH	Real	Mach number
BCID	Integer	Boundary condition ID
SUB	Integer	Trim condition subscript
BASEPARM	Text(8)	Character string identifying the basis configuration condition in AIRFPRM.
BASEVAL	Real	Parameter value in basis configuration
PARM1	Text(8)	Character string identifying the configuration parameter
PARM1VAL	Real	Delta parameter value used to generate the "unit" forces
PARM2	Text(8)	Character string identifying the configuration parameter
PARM2VAL	Real	Delta parameter value used to generate the "unit" forces
PARM3	Text(8)	Character string identifying the configuration parameter
PARM3VAL	Real	Delta parameter value used to generate the "unit" forces
SYMFLAG	Integer	Symmetry flag for the parameter
QDP	Real	Dynamic Pressure (0.0 if Rigid and > 0.0 if Flexible)
SCFTYPE	Text(8)	Character string identifying stability coefficient definition
CL	Real	Lift Coefficient
CD	Real	Drag Coefficient
CS	Real	Side Force Coefficient
CMX	Real	Rolling Moment Coefficient
CMY	Real	Pitching Moment Coefficient
CMZ	Real	Yawing Moment Coefficient

Created By: Module RGDSTB

Notes:

1. The SYMFLAG values are:
   - = 1 Symmetric
   - = 0 Asymmetric
   - = -1 Antisymmetric

2. (All terms in the relation will either be asymmetric or symmetric and antisymmetric. No mix of asymmetric with antisymmetric or symmetric is allowed.)

3. The QDP values are:

   = 0.0 If Rigid

   > 0.0 If Flexible

4. The SCFTYPE values are:

   = RIGID

5. BASEPARM identifies the basis configuration load set. It replaces the THKCAM parameter in the previous ASTROS paradigm, and must be known to provide point of reference for the stability derivatives as increments from the basis configuration.

6. PARM identifies the physical variable whose perturbation generated the rigid coefficients. There are six accelerations and six configuration parameters whose names are reserved that have special meaning. Additional PARM values come from the AESURF control surfaces and SLPARM user-defined control effectors. The PARM attribute contains the user supplied label. For given BCID and SUB values, the SAROLOAD matrix has one column (which may contain only zeros) for each entry of the STABCFA relation. The PARM field has 3 possible entries to accommodate future nonlinear loads. The first PARM, PARM1 is the primary parameter, while the other parameters are dependent on PARM1. The PARM field is then:

PARM	VARIABLE
NX	Rigid body acceleration in drag/thrust direction (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.)
NY	Rigid body acceleration in side force direction (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.)
NZ	Rigid body acceleration in plunge direction (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.)
PACCEL	Rigid body acceleration about the roll axis. (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.)
QACCEL	Rigid body acceleration about the pitch axis. (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.)
RACCEL	Rigid body acceleration about the yaw axis. (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.)
ALPHA	Forces arising due to incremental angle of attack from the basic aerodynamic configuration.
BETA	Forces arising due to incremental angle beta from the basic aerodynamic configuration.
PRATE	Forces arising due to unit roll rate
QRATE	Forces arising due to unit pitch rate
RRATE	Forces arising due to unit yaw rate

SURFACE	Forces arising due to incremental deflection from the basic aerodynamic configuration of the AESURF control surface named in the PARM field.
UDEFCONTROLLER	Forces arising from the SLPARM control effector named in the PARM field. (Produces no forces in the aerodynamic domain, but is included for completeness to allow modification of the UDFALOAD

Entity: **STABCFS**

Entity Type: Relation

Description: Stability Coefficients In Structural Domain.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MODELNAME	Text(8)	Name of steady aerodynamic input packet
METHOD	Text(8)	Name of method which created load vector
MACH	Real	Mach number
BCID	Integer	Boundary condition ID
SUB	Integer	Trim condition subscript
BASEPARM	Text(8)	Character string identifying the basis configuration condition in FLEXPRM.
BASEVAL	Real	Parameter value in basis configuration
PARM1	Text(8)	Character string identifying the configuration parameter
PARM1VAL	Real	Delta parameter value used to generate the "unit" forces
PARM2	Text(8)	Character string identifying the configuration parameter
PARM2VAL	Real	Delta parameter value used to generate the "unit" forces
PARM3	Text(8)	Character string identifying the configuration parameter
PARM3VAL	Real	Delta parameter value used to generate the "unit" forces
SYMFLAG	Integer	Symmetry flag for the parameter
QDP	Real	Dynamic Pressure (0.0 if Rigid and > 0.0 if Flexible)
SCFTYPE	Text(8)	Character string identifying stability coefficient definition
CL	Real	Lift Coefficient
CD	Real	Drag Coefficient
CS	Real	Side Force Coefficient
CMX	Real	Rolling Moment Coefficient
CMY	Real	Pitching Moment Coefficient
CMZ	Real	Yawing Moment Coefficient

Created By: Module FLXSTB

Notes:

1. The SYMFLAG values are:
   - = 1 Symmetric
   - = 0 Asymmetric
   - = -1 Antisymmetric

(All terms in the relation will either be asymmetric, or symmetric and antisymmetric. No mix of asymmetric with antisymmetric or symmetric is allowed.)

1. The QDP values are:
   - = 0.0 If Rigid
   - > 0.0 If Flexible

2. The SCFTYPE values are:
   = RSPLINED
   = RESTRAIN
   = UNRESTRA
1. BASEPARM identifies the basis configuration load set. It replaces the THKCAM parameter in the previous ASTROS paradigm, and must be known to provide point of reference for the stability derivatives as increments from the basis configuration.
2. PARM identifies the physical variable whose perturbation generated the rigid coefficients. There are six accelerations and six configuration parameters whose names are reserved that have special meaning. Additional PARM values come from the AESURF control surfaces and SLPARM user-defined control effectors. The PARM attribute contains the uses supplied label. For given a BCID and SUB combination, the LHSA and RHSA matrices have one column for each entry of the STABCFS relation. The PARM field has 3 possible entries to accommodate future nonlinear loads. The first PARM, PARM1 is the primary parameter, while the other parameters are dependent on PARM1. The PARM field is then:

PARM	VARIABLE
NX	Aeroelastic response to rigid body acceleration in drag/thrust direction. (These terms are nonzero only for restrained derivatives.)
NY	Aeroelastic response to rigid body acceleration in side force direction. (These terms are nonzero only for restrained derivatives.)
NZ	Aeroelastic response to rigid body acceleration in plunge direction. (These terms are nonzero only for restrained derivatives.)
PACCEL	Aeroelastic response to rigid body acceleration about the roll axis. (These terms are nonzero only for restrained derivatives.)
QACCEL	Aeroelastic response to rigid body acceleration about the pitch axis. (These terms are nonzero only for restrained derivatives.)
RACCEL	Aeroelastic response to rigid body acceleration about the yaw axis. (These terms are nonzero only for restrained derivatives.)
ALPHA	Forces arising due to incremental angle of attack from the basic aerodynamic configuration.
BETA	Forces arising due to incremental angle beta from the basic aerodynamic configuration.
PRATE	Forces arising due to unit roll rate
QRATE	Forces arising due to unit pitch rate
RRATE	Forces arising due to unit yaw rate
SURFACE	Forces arising due to incremental deflection from the basic aerodynamic configuration of the AESURF control surface named in the PARM field.
UDEFCONTROLLER	Forces arising the SLPARM control effector named in the PARM field. (Produces no forces in the aerodynamic domain, but is included for completeness to allow modification of the UDFFORCE.)

Entity: **STDYGEOM**

Entity Type: Relation

Description: Steady aerodynamic model geometry group

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	C(24)	Formal member name
MEMBERCL	C(8)	Entity classification (i.e. RELATION, MATRIX, UNSTRUCTured)
ENTITYNA	C(52)	Name of entity and logical database where it resides
DATE	C(8)	Date when MEMBER was added to this group
TIME	C(8)	Time when MEMBER was added to this group
MODELNAM	C(8)	Name of steady aerodynamic model
MODELTYP	C(8)	Model symmetry type (SYM or ASYM)
AEROMETH	C(8)	Name of method which created aerodynamic geometry
TITLE	C(72)	User label TITLE
SUBTITLE	C(72)	User label SUBTITLE
LABEL	C(72)	User label LABEL

Created By: Module AROGNDRV

Notes:

1. If the solution sequence does not specify model assembly commands, indexed entity names are assigned by default (MEMBERNA with the appended index), otherwise entity names are defined as specified by the model assembly commands. (Entity name indices are independent of other members.)

2. Entity names can be shared by multiple groups.

3. MEMBERNA must be unique within a specific group unless otherwise specified.

Entity: **TCONBMST**

Entity Type: Relation

Description: Contains the component load (BMST) constraints for the trim optimization problem

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	Integer > 0	Set identification number
BMSTLAB	Text(8)	Component load (BMST) label
CMPNT	Integer	Component number (DOF) of BMST (+/- 1, 2, 3, 4, 5, or 6)
CTYPE	Text(8)	Constraint type (UPPER or LOWER)
BMSTLIM	Real	Component load bound

Created By: Module IFP

Notes:

1. A negative **CMPNT** refers to the image side when substructuring.

Entity: **TCONFUNC**

Entity Type: Relation

Description: Contains the functional constraints for the trim optimization problem

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SID	Integer > 0	Set identification number
TFID	Integer > 0	Trim optimization function identification number
CTYPE	Text(8)	Constraint type (UPPER or LOWER)
FUNCLIM	Real	Function value bound

Created By: Module IFP

Entity: **TCONTRM**

Entity Type: Relation

Description: Contains the trim parameter constraints for the trim optimization problem

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	Integer > 0	Set identification number
PRMLAB	Text(8)	Constrained control surface label or aeroelastic trim parameter
CTYPE	Text(8)	Constraint type (UPPER or LOWER)
PRMLIM	Real	Trim parameter bound

Created By: Module IFP

Notes:

1. Refer to the **TCONTRM** Bulk Data entry for further information.

Entity: **TFUNC**

Entity Type: Relation

Description: Contains weighted sum function definitions to be used in trim optimization

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
TFID	Integer > 0	Trim optimization function identification number
FTYPEI	Text(4)	Function argument type (PARM, BMST, or DRAG)
FNAMEI	Text(8)	Function argument name
FIDI	Integer > 0	Function argument identification number
WFACTI	Real	Weighting factor for this function argument

Created By: Module IFP

Notes:

1. Refer to the **TFUNC** Bulk Data Entry for further information.

Entity: **TLABEL**

Entity Type: Unstructured

Description: Master label list for development of TRIMTOC and Loads for steady aeroelastic analysis

Entity Structure: Record 1 contains one integer defining the number of trim labels (NLABEL) required for the current boundary condition and subscript trim sets.
Record 2 contains an array of 3 logical variables defining the required aerodynamic symmetries to build the trim cases in the current boundary condition and subscript.
Record 3 contains the NLABEL tuples of trim label (2 words) and trim parameter symmetry (1 word)

Created By: Module ACCFGEN

Entity: **TODVPRM**

Entity Type: Relation

Description: Contains the trim optimization design variables

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SID	Integer > 0	Set identification number
DVID	Integer > 0	Design variable identification number
PRMLAB	Text(8)	Trim parameter label
DVMIN	Real	Minimum allowable value of the design variable
DVMAX	Real	Maximum allowable value of the design variable
DVINIT	Real	Initial value of the design variable
DVTRIM	Text(4)	Character string TRIM if initial value determined by trim solution

Created By: Module IFP

Entity: **TOMPPARM**

Entity Type: Relation

Description: Contains the optimizer parameters and their new values for use in the trim mathematical programming optimizer

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
TRIMID	Integer > 0	Trim identification number
PARAM	Text(8)	Name of parameter to be overridden
INTPARM	Integer	Value of integer parameters
RSPPARM	Real	Value of real parameters

Created By: Module IFP

Notes:

1. This relation is used in the trim optimization module to provide for user specification of optimizer parameters.

Entity: **TRIM**

Entity Type: Relation

Description: Contains the specified conditions for steady aeroelastic trim as input from Bulk Data file.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	Integer > 0	Trim set identification number
MACH	Real > 0.0	Mach number
QDP	Real > 0.0	Dynamic pressure
TRMTYP	Text(8)	Type of trim desired
EFFID	Integer	Identification of CONEFFS bulk data entries which modify control surface effectiveness values
V0	Real	Freestream velocity
LABELI	Text(8)	Label defining the aerodynamic trim parameters
FREEI	Text(4)	Blank, character string FREE, or character string SCHD
FIXI	Real	Magnitude of the trim parameter

Created By: Module IFP

Entity: **TRIMDATA**

Entity Type: Relation

Description: Contains the TRIM Bulk Data and related boundary condition, subcase, and subscript information

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	Integer > 0	Trim set identification number
MACHINDX	Integer	Mach number index for the current subcase
SYMMFLAG	Integer	Symmetry option for the Mach number
SUBSCRPT	Integer	Subscript counter
SUBCASID	Integer	Subcase identification number
BCID	Integer	Boundary condition identification number
METHOD	Text(8)	Name of method which created aerodynamic data
MODELNAM	Text(8)	Name of steady aerodynamic input packet
MACH	Real > 0.0	Mach number
QDP	Real > 0.0	Dynamic pressure
TRMTYP	Text(8)	Type of trim desired (LIFT, PITCH, ROLL, or blank)
EFFID	Integer	Identification of CONEFFS bulk data entries which modify control surface effectiveness values
V0	Real	Freestream velocity
LABELI	Text(8)	Label defining the aerodynamic trim parameter
FREEI	Text(4)	Character string FREE, SCHD, or blank
FIXI	Real	Magnitude of the trim parameter

Created By: Module TRIMCHEK

Entity: **TRIMOPT**

Entity Type: Relation

Description: Contains the trim optimization problem specification

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
TRIMID	Integer > 0	Trim set identification number of the associated TRIM or RTRIM entry
OPTFLG	Text(4)	Optimization flag (MIN or MAX)
OBJTYP	Text(4)	Objective function type (FUNC, PARM, BMST or DRAG)
OBJNAM	Text(8)	Objective function name
OBJID	Integer	Objective function identification number or BMST load component number
TCONID	Integer > 0	Trim optimization constraint set identification number
TDVSID	Integer > 0	Trim optimization design variable set identification number
MAXITER	Integer > 0	Maximum number of optimization iterations to be performed
MOVLIM	Real > 0.0	Move limit applied to trim optimization design variables
CNVGLIM	Real > 0.0	Convergence limit specifying the maximum imbalance in a supported degree of freedom that can be considered converged
PRINT	Integer > 0	Trim optimization μ-**DOT** print flag

Created By: Module IFP

Notes:

1. Refer to the TRIMOPT Bulk Data Entry for further information.

Entity: **TRIMR**

Entity Type: Relation

Description: Contains the specified conditions for steady rigid aerodynamic trim as input from Bulk Data file.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
SETID	Integer > 0	Trim set identification number
MACH	Real > 0.0	Mach number
QDP	Real > 0.0	Dynamic pressure
TRMTYP	Text(8)	Type of trim desired
EFFID	Integer	Identification of CONEFFS bulk data entries which modify control surface effectiveness values
V0	Real	Freestream velocity
XREF	Real	Reference x-axis location for sum of moments
YREF	Real	Reference y-axis location for sum of moments
ZREF	Real	Reference z-axis location for sum of moments
MASSID	Integer > 0	Identification number of RMASS Bulk Data entry specifying the inertia data to be used for trim
CMPNTS	Integer	Component number(s) to be used for trim
LABELI	Text(8)	Label defining the aerodynamic trim parameters
FREEI	Text(4)	Blank, character string FREE, or character string SCHD
FIXI	Real	Magnitude of the trim parameter

Created By: Module IFP

Entity: **TRIMRSLT**

Entity Type: Relation

Description: Contains the TRIM Bulk Data, related boundary condition, subcase, and subscript information, and the results of the trim solutions.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
TRIMID	Integer > 0	Trim set identification number
MACH	Real > 0.0	Mach number
QDP	Real > 0.0	Dynamic pressure
TRMTYP	Text(8)	Type of trim desired (LIFT, PITCH, ROLL, or blank)
EFFID	Integer	Identification of CONEFFS bulk data entries which modify control surface effectiveness values
V0	Real	Freestream velocity
LABELI	Text(8)	Label defining the aerodynamic trim parameter
PRMTYPI	Text(4)	Type of trim parameter (FREE, SCHD, or blank)
VALUEI	Real	Value of trim parameter after trim solution
MACHINDX	Integer	Mach number index for the current subcase
SYMMFLAG	Integer	Symmetry option for the Mach number
SUBSCRPT	Integer	Subscript counter
SUBCASID	Integer	Subcase identification number
BCID	Integer	Boundary condition identification number
SOURCE	Real(4)	Source of VALUEI (INIT, TRIM, or TOPT)

Created By: Modules PRETRM, SAERODRV, FLEXTRIM, RIGDTRIM, FTRIMOPT, and RTRIMOPT

Notes:

1. The **TRIMRSLT** relation is initialized in module **PRETRM** with a copy of the **TRIMDATA** relation and updated after each trim solution.

Entity: **TRIMTOC**

Entity Type: Relation

Description: Trim Parameter Table of Contents - Pointer to PAF.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MEMBERNA	Text(24)	Formal Member Name (RIGDALOD, RIGDSLOD)
MEMBERCL	Text(8)	Entity Classification (i.e., RELATION, MATRIX, UNSTRUCTured, or GROUP)
ENTITYNAM	Text(52)	Name of entity and logical database where it resides
MODELNAM	Text(8)	Name of steady aerodynamic geometry packet
METHOD	Text(8)	Name of method which created aerodynamic geometry
MACH	Real	Mach Number of Aerodynamic Data
V0	Real	Freestream velocity used to dimensionalize rate terms
SYMFLAG	Integer	Symmetry flag for the parameter
PARMTYP	Text(8)	Parameter type (i.e. acceleration, static load parameter, or aerodynamic parameter)
PARAM1	Text(8)	Character string identifying first level trim parameter in RIGDALOD or RIGDSLOD group.
PARAM1VA	Real	Delta parameter value used to generate the "unit" forces
PARAM2	Text(8)	Character string identifying second level trim parameter in RIGDALOD or RIGDSLOD group.
PARAM2VA	Real	Delta parameter value used to generate the "unit" forces
PARAM3	Text(8)	Character string identifying third level trim parameter in RIGDALOD or RIGDSLOD group.
PARAM3VA	Real	Delta parameter value used to generate the "unit" forces
COLNUM	Integer	Column number in Load Matrix

Created By: Modules ACCFGEN, UDEFGEN, and ARFMRG

Notes:

1. TRIMTOC is indexed on boundary condition and subscript (Model, Method, Mach).

2. The SYMFLAG values are:
   - = 1 Symmetric
   - = 0 Asymmetric
   - = -1 Antisymmetric

   (All terms in the relation will either be asymmetric or, symmetric and antisymmetric. No mix of asymmetric with antisymmetric or symmetric is allowed.)

3. The PARMTYP values are:

= ACCE	Acceleration parameter (e.g. NZ, PACCEL)
= STRC	Static load parameter (SPLARM)
= AERO	Aerodynamic parameter
= BASE	Aerodynamic basis parameter

PARAMs	VARIABLE
PARAMREF (default name)	Nominal reference configuration comprising a combination of onset flow and trim parameter deflections from a subset of all the trim parameters. (See Note 1)
NX (default name)	Rigid body acceleration in drag/thrust direction (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.) (See Note 1)
NY (default name)	Rigid body acceleration in side force direction (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.) (See Note 1)
NZ (default name)	Rigid body acceleration in plunge direction (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.) (See Note 1)
PACCEL (default name)	Rigid body acceleration about the roll axis. (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.) (See Note 1)
QACCEL (default name)	Rigid body acceleration about the pitch axis. (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.) (See Note 1)
RACCEL (default name)	Rigid body acceleration about the yaw axis. (Produces no forces, but included for completeness to provide for restrained derivative computations in aeroelastic solutions.) (See Note 1)
ALPHA (default name)	Forces arising due to incremental angle of attack from the basic aerodynamic configuration. (See Note 1)
BETA (default name)	Forces arising due to incremental angle beta from the basic aerodynamic configuration. (See Note 1)
PRATE	Forces arising due to unit roll rate
QRATE	Forces arising due to unit pitch rate
RRATE	Forces arising due to unit yaw rate
SURFACE	Forces arising due to incremental deflection from the basic aerodynamic configuration of the AESURF control surface named in the PARAM field.
UDEFCONTROLLER	Forces arising from the SLPARM control effector named in the PARAM field. (Produces no forces in the aerodynamic domain, but is included for completeness to allow modification of the SLPFRC.)

Notes:

1. ASTROS default names for parameters are the same as the original paradigm. In the case of Alternate Aerodynamic Database, the user may name these characteristic parameters any name.

Entity: **UDFALOAD**

Entity Type: Matrix

Description: Matrix containing the null columns for user defined loads in the aerodynamic domain.

Matrix Form: Rectangular and real. The number of rows is equal to the number of panels in the steady aerodynamics model and the number of columns is equal to the number of user-defined load (STRC) parameters in TRIMTOC.

Created By: Module UDEFGEN

Notes:

1. User defined loads can only be defined in the structural domain.
2. Refer to UDEFGEN module documentation for more information.

Entity: **UDFFORCE**

Entity Type: Matrix

Description: Matrix containing the columns for user defined loads in the structural domain in the n-set

Matrix Form: Rectangular and real. The number of rows is equal to the number of degrees-of-freedom in the f-set and the number of columns is equal to the number of user-defined load (STRC) parameters in TRIMTOC.

Created By: MAPOL

Entity: **UDFFORCX**

Entity Type: Matrix

Description: Matrix containing the columns for user defined loads in the structural domain in the f-set for the full structure

Matrix Form: Rectangular and real. The number of rows is equal to the number of degrees-of-freedom in the f-set and the number of columns is equal to the number of user-defined load (STRC) parameters in TRIMTOC.

Created By: Module UDEFTRAN (if substructuring) or MAPOL (if not substructuring)

Notes:

1. UDFFORCX is set equal to UDFFORCE in MAPOL if not substructuring.
2. Refer to UDEFTRAN module documentation for more information.

Entity: **UDGFORCE**

Entity Type: Matrix

Description: Matrix containing the columns for user defined loads in the structural domain in the g-set

Matrix Form: Rectangular and real. The number of rows is equal to the number of degrees-of-freedom in the g-set and the number of columns is equal to the number of user-defined load (STRC) parameters in TRIMTOC.

Created By: Module UDEFGEN

Notes:

1. Refer to UDEFGEN module documentation for more information.

Entity:	**UDNFORCE**
Entity Type:	Matrix
Description:	Matrix containing the columns for user defined loads in the structural domain in the n-set
Matrix Form:	Rectangular and real. The number of rows is equal to the number of degrees-of-freedom in the n-set and the number of columns is equal to the number of user-defined load (STRC) parameters in TRIMTOC.
Created By:	MAPOL

THIS PAGE INTENTIONALLY LEFT BLANK.

5. REFERENCES

1. Johnson, E.H. and Venkayya, V.B. "Automated Structural Optimization System (ASTROS), Theoretical Manual, "AFWAL-TR-88-3028, Vol. 1 December 1988.

2. Love, M.H., “Software Design Document for The Aerodynamic Analysis for the Design Environment,” FZM 8399, 17 June 1996.

3. Love, M.H., “Aerodynamic Analysis for the Design Environment Final Report Theoretical and Application Studies Document,” FZM 8536, 31 July 1998.

4. Love, M.H., “Aerodynamic Analysis for the Design Environment User’s Document,” FZM 8538, 24 July 1998.

5. Neill, D.J., Herendeen, D.L., Venkayya, V.B., “ASTROS Enhancements, Volume II - ASTROS Programmer’s Manual,” WL-TR-95-3006, May 1995.